APPROXIMATION SCHEMES TO SIMPLIFY POSTERIOR COMPUTATION

A Dissertation

by

ALLYSON E. LARSEN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Anirban Bhattacharya
Committee Members,	Irina Gaynanova
	Bani Mallick
	Xiaoning Qian
Head of Department,	Daren Cline

August 2020

Major Subject: Statistics

ABSTRACT

Markov chain Monte Carlo (MCMC) sampling methods often do not scale well to large datasets, so there has been an increased interest in approximate Markov chain Monte Carlo (aMCMC) sampling methods. We propose two different aMCMC methods. For the first method, we propose a new distribution, called the soft tMVN distribution, which provides a smooth approximation to the truncated multivariate normal (tMVN) distribution with linear constraints. The soft tMVN distribution can be used to approximate simulations from a multivariate truncated normal distribution with linear constraints, or itself as a prior in shape-constrained problems. We provide theoretical support to the approximation capability of the soft tMVN and provide further empirical evidence thereof. We then develop an aMCMC method for Bayesian monotone single-index modeling. We replace the usual tMVN prior with the soft tMVN prior and show that using the soft tMVN prior gives similar statistical performance while the run-time is significantly faster.

The second aMCMC method is a multivariate convex regression method. In it, we approximate the max of affine functions with the softmax of affine functions. Convex regression methods that use the max of affine functions appear to do well in traditional frequentist settings, but does not scale well to large data in Bayesian settings. We propose the softmax-affine convex (SMA) regression method which replaces the max with the softmax function. The softmax function is a smooth function that approximates the max of affine functions. This allows gradients to be computed, which makes the Hamiltonian Monte Carlo (HMC) algorithm a natural choice for sampling from the posterior. We specify the priors for SMA and use Stan, a default HMC algorithm, to sample from the posterior. We provide empirical evidence that SMA regression is comparable to existing convex regression methods. We also provide a method for choosing the number of affine functions in the softmax function.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Anirban Bhattacharya [advisor], Irina Gaynanova, and Bani Mallick of the Department of Statistics and Professor Xiaoning Qian of the Department of Electrical and Computer Engineering.

All work for the dissertation was completed by the student, in collaboration with Anirban Bhattacharya of the Department of Statistics. Additionally, Sections 2 and 3 were done in collaboration with Debdeep Pati of the Department of Statistics and were posted on arXiv.

Funding Sources

Graduate study was supported by a fellowship from Texas A&M University.

NOMENCLATURE

aMCMC	approximate Markov chain Monte Carlo
ARS	adaptive rejection sampling
CAP	Convex Adaptive Partitioning
GP	Gaussian Process
HMC	Hamiltonian Monte Carlo
LMC	Langevin Monte Carlo
LSE	Least Squares Estimator
MALA	Metropolis adjusted Langevin
MBCR	Multivariate Bayesian Convex Regression
MCMC	Markov chain Monte Carlo
NB	Negative Binomial distribution
n.n.d	non-negative definite matrix
p.d.	positive definite matrix
PG	Polya-Gamma distribution
RJMCMC	Reversible Jump Markov chain Monte Carlo
SMA	Softmax-affine convex regression
tMVN	truncated multivariate normal distribution

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Fully Bayesian modeling approaches are desirable, because they allow for quantifications of uncertainty. However, they often require sampling using Markov chain Monte Carlo (MCMC) algorithms, which do not scale well to large datasets. Therefore, there has recently been an increased interest in approximate Markov chain Monte Carlo (aMCMC) algorithms [Johndrow et al., 2015, Bardenet et al., 2017], which are methods where the exact transition kernel of a Markov chain is replaced by an approximation for computational ease.

One aMCMC method is the subsampling method [Bardenet et al., 2017, Welling and Teh, 2011, Quiroz et al., 2019, Bardenet et al., 2014, Korattikara et al., 2014], which only uses a subsample of the data at each likelihood evaluation inside the MCMC algorithm. There are theoretical results on the convergence of these aMCMC methods [Pillai and Smith, 2014, Quiroz et al., 2019, Mitrophanov, 2005, Rudolf et al., 2018], that give bounds on the bias [Quiroz et al., 2019], and that test when aMCMC algorithms are "better" than MCMC methods [Johndrow et al., 2015].

Another aMCMC method is the divide-and-conquer method [Bardenet et al., 2017, Payne and Mallick, 2018]. This method splits the data into batches, fits an MCMC algorithm on each batch separately, then combines the batch posterior estimates into the final approximate posterior estimate. Batch posterior estimates can be combined into the final posterior estimate using Gaussian approximations or importance sampling [Huang and Gelman, 2005], using averaging [Scott et al., 2016], by multiplying smooth approximations to the batch posteriors [Neiswanger et al., 2013], or by using a Weierstrass sampler [Wang and Dunson, 2013]. These methods are often only theoretically sound when the batch posteriors are Gaussian or when the batch sizes approach infinity.

A third aMCMC method is to approximate the likelihood, a prior distribution, or some other distribution with a different, more computationally feasible distribution. For example, using a low-rank covariance approximation in Gaussian process models [Johndrow et al., 2015], approximating the multivariate logistic distribution using the t distribution [O'brien and Dunson, 2004], approximating full conditionals by beta densities [Bhattacharya and Dunson, 2012], and thresholding in

sparse models [Johndrow et al., 2017].

The approximation schemes developed in this dissertation all fall under the third aMCMC approach. Section 2 introduces the soft truncated multivariate normal distribution, which approximates the truncated multivariate normal distribution. Theoretical support and empirical evidence to the approximation capability of the soft tMVN distribution are provided. An efficient blocked Gibbs sampler is developed to sample from the soft tMVN distribution. Section 3 contains an application of the soft tMVN distribution to a monotone single-index model. An aMCMC algorithm which replaces the tMVN distribution with the soft tMVN distribution as a prior is developed. Both Section 2 and Section 3 have been posted on arXiv [Souris et al., 2018]. Section 4 develops a softmax-affine convex (SMA) regression method. The SMA regression method is based on a convex regression method [Hannah and Dunson, 2011, 2013] that uses the maximum of affine functions. We approximate the maximum of affine functions with a soft version of the maximum. Section 5 contains other work that I completed during my doctorate studies, but does not relate to approximate MCMC methods. In it, we explore the covariance structure of factor models for multivariate count data. The covariance of a Gaussian factor model decomposes into the sum of a low rank matrix and diagonal matrix. We show both theoretically and numerically that while the covariance of count data factor models does not decompose into the sum of a low rank matrix and diagonal matrix, it does decompose into the sum of a low effective rank matrix and diagonal matrix. All sections (except the conclusion, Section 6) contain their own introduction and discussion subsections.

2. THE SOFT MULTIVARIATE TRUNCATED NORMAL DISTRIBUTION

2.1 Introduction

The problem of sampling from a truncated multivariate normal (tMVN) distribution with linear constraints is frequently encountered as a component of a larger Markov chain Monte Carlo (MCMC) algorithm to sample from the full conditional distribution of a constrained parameter vector. As a running example revisited on multiple occasions in this section, consider binary variables $y_i = \mathbb{1}(z_i > 0)$, with $z = (z_1, \ldots, z_n)^{\mathrm{T}}$ a vector of latent Gaussian thresholds [Albert and Chib, 1993] and $w \in \mathbb{R}^q$ a vector of parameters/latent variables so that the joint distribution of $\theta = (z, w)$ follows a $\mathcal{N}(\mu, \Sigma)$ distribution. It then immediately follows that the (conditional) posterior of $\theta \mid y, \mu, \Sigma$ follows a $\mathcal{N}(\mu, \Sigma)$ distribution truncated to $\otimes_{i=1}^{n} \mathcal{C}_i \otimes \mathbb{R}^q$, with $\mathcal{C}_i = (0, \infty)$ or $(-\infty, 0)$ depending on whether $y_i = 1$ or 0. Such latent Gaussian threshold models are ubiquitous in the analysis of binary and nominal data; examples include probit regression and its multivariate extensions [Albert and Chib, 1993, Holmes et al., 2006, Chib and Greenberg, 1998, O'brien and Dunson, 2004], multinomial probit models [McCulloch et al., 2000, Zhang et al., 2008, Johndrow et al., 2013], tobit models [Tobin, 1958, Polasek and Krause, 1994], and binary Gaussian process (GP) classification models [Girolami and Rogers, 2006] among others.

In this section, we propose a new family of distributions called the soft tMVN distribution which replaces the hard constraints in a tMVN distribution with a smoothed or "soft" version using a logistic sigmoid function. The soft tMVN distribution admits a smooth log-concave density on the d-dimensional Euclidean space. Although the soft tMVN distribution is supported on the entire d-dimensional space, it can be made to increasingly concentrate most of its mass on a polyhedron determined by multiple linear inequality constraints, by tweaking a parameter. In fact, we show that the soft tMVN distribution approximates the corresponding tMVN distribution in total variation distance.

Recognizing the soft tMVN distribution as the posterior distribution in a pseudo-logistic re-

gression model, we develop an efficient blocked Gibbs sampler combining the Polya–Gamma data augmentation of Polson et al. [2013] along with a structured multivariate normal sampler from Bhattacharya et al. [2016]. In contrast, existing Gibbs samplers for a tMVN distribution sample the coordinates one-at-a-time from their respective full conditional univariate truncated normal distributions [Geweke, 1991, Kotecha and Djuric, 1999, Damien and Walker, 2001, Rodriguez-Yam et al., 2004]. The algorithm of Geweke is implemented in the `R` package `tmvtnorm` [Wilhelm and G, 2015]. While the Gibbs sampling procedure is entirely automated, it is well-recognized in a broader context that such one-at-a-time updates can lead to slow mixing, especially if the variables are highly correlated. We have additionally observed numerical instabilities in the `R` implementation for unconstrained dimensions exceeding 400. While exact Hamiltonian Markov chain (HMC) algorithms to sample from tMVN [Pakman and Paninski, 2014] are also popular, such algorithms are not suitable to sample from the soft tMVN, and leaf-frog steps with careful tuning are necessary to obtain good mixing. There also exists accept-reject algorithms for the tMVN distribution that create exact samples from the distribution [Botev, 2017]. The algorithm of Botev is implemented in the `R` package `TruncatedNormal` [Botev, 2015]. While exact samples are possible, when the acceptance probability becomes small, either the algorithm slows tremendously or approximate samples are produced. We typically saw small acceptance probabilities in the `R` implementation when the constrained dimension exceeded 200. With such motivation, we propose to replace a tMVN distribution with its softened version inside a larger MCMC algorithm and use our sampling strategy for the soft tMVN distribution. In recent years, there has been several instances of such approximate MCMC (aMCMC) [Johndrow et al., 2015] algorithms where the exact transition kernel of a Markov chain is replaced by an approximation thereof for computational ease.

Like the tMVN distribution, the soft tMVN distribution is conditionally conjugate for the mean in a Gaussian likelihood. The soft tMVN can be viewed as a shrinkage prior which encourages shrinkage towards a linearly constrained region rather than being supported on the region. There is an interesting parallel between the soft tMVN distribution and global-local shrinkage priors used in sparse regression problems. The global-local priors replace the point mass (at zero) of the more

traditional discrete mixture priors and rather encourage shrinkage towards the origin, with the motivation that a subset of the regression coefficients may have a small but non-negligible effect. Similarly, the soft tMVN prior favors the shape constraints while allowing for small departures.

The rest of the section is organized as follows. In subsection 2.2, we introduce the soft tMVN distribution as an approximation to the tMVN distribution and discuss its properties. In subsection 2.3, we discuss various strategies to sample from a soft tMVN distribution, including a scalable Gibbs sampler suitable for high-dimensional situations. Subsection 2.4 contains a number of simulation examples to illustrate the efficacy of the proposed sampler as well as the approximation capability of the soft tMVN distribution. We conclude with a discussion in Subsection 2.5.

2.2 The soft tMVN distribution

Consider a tMVN distribution

$$\gamma(\theta) \propto e^{-\frac{1}{2}(\theta-\mu)^{\mathrm{T}}\Sigma^{-1}(\theta-\mu)}\, \mathbb{1}_{\mathcal{C}}(\theta), \tag{2.1}$$

where $\mu \in \mathbb{R}^d$, Σ is a $d \times d$ positive definite matrix, and $\mathcal{C}$ is described by $r \leq d$ linear constraints,

$$\mathcal{C} = \left\{ \theta \in \mathbb{R}^d : s_i\,(a_i^{\mathrm{T}}\theta) \geq 0,\ i = 1, \ldots, r \right\},$$

where $s_i \in \{1, -1\}$ denotes the sign of the ith inequality, and $a_i \in \mathbb{R}^d$. Without loss of generality, we assume the first r coordinates to be constrained; this is mainly for notational convenience and can always be achieved by reordering the variables, if necessary. We also assume throughout that $\mathcal{C}$ has positive $\mathbb{R}^d$-Lebesgue measure, so that the density γ in (2.1) is non-singular on $\mathbb{R}^d$. In the special case where $a_i = e_i$, the ith unit vector in $\mathbb{R}^d$ (with 1 at the ith coordinate and 0 elsewhere), the constraint set $\mathcal{C}$ reduces to the form $\otimes_{i=1}^{r}\mathcal{C}_i \otimes \mathbb{R}^q$ mentioned in the introduction. While this is an important motivating example, our approach works more generally for the type of constraints in the above display.

Write, using the convention $0^0 = 1$,

$$\begin{aligned}\mathbb{1}(\theta \in \mathcal{C}) &= \prod_{i\in[r]\,:\,s_i=1} \mathbb{1}(a_i^{\mathrm{T}}\theta \geq 0) \prod_{i\in[r]\,:\,s_i=-1} \mathbb{1}(a_i^{\mathrm{T}}\theta < 0) \\ &= \prod_{i=1}^{r} \{\mathbb{1}(a_i^{\mathrm{T}}\theta \geq 0)\}^{\mathbb{1}(s_i=1)} \{\mathbb{1}(a_i^{\mathrm{T}}\theta < 0)\}^{\mathbb{1}(s_i=-1)}.\end{aligned}$$

Our main idea is to replace the indicator functions above with a smoothed or "soft" approximation. A rich class of approximations to the indicator function $\mathbb{1}_{(0,\infty)}(\cdot)$ is provided by sigmoid functions, which are non-negative, monotone increasing, differentiable, and satisfy $\lim_{x\to\infty} \sigma(x) = 1$ and $\lim_{x\to-\infty} \sigma(x) = 0$. The cumulative distribution function of any absolutely continuous distribution on $\mathbb{R}$ which is symmetric about zero can be potentially used as a sigmoid function. Here, for reasons to be apparent shortly, we choose to use the logistic sigmoid function $\sigma(x) = 1/(1+e^{-x})$, which is the cdf of the logistic distribution. Specifically, define, for $\eta > 0$,

$$\sigma_\eta(x) = \frac{1}{1+e^{-\eta x}} = \frac{e^{\eta x}}{1+e^{\eta x}}, \quad x \in \mathbb{R}, \tag{2.2}$$

to be a scaled version of $\sigma(\cdot)$. The parameter η controls the quality of the approximation, with larger values of η providing increasingly better approximations to $\mathbb{1}_{(0,\infty)}(\cdot)$. In fact, it is straightforward to see that

$$|\sigma_\eta(x) - \mathbb{1}_{(0,\infty)}(x)| \leq \frac{1}{1+e^{\eta|x|}}, \quad x \in \mathbb{R}. \tag{2.3}$$

It is also immediate that $(1-\sigma_\eta(\cdot))$ is an approximation to $\mathbb{1}_{(-\infty,0)}(\cdot)$ with the same approximation error.

We are now ready to describe our approximation scheme. Fixing some large η and replacing the indicators by their respective sigmoidal approximations in (2.1), we obtain the approximation

γ_η to γ as

$$\gamma_\eta(\theta) \propto e^{-\frac{1}{2}(\theta-\mu)^{\mathrm{T}}\Sigma^{-1}(\theta-\mu)} \prod_{i=1}^{r} \left(\frac{e^{\eta\, a_i^{\mathrm{T}}\theta}}{1+e^{\eta\, a_i^{\mathrm{T}}\theta}}\right)^{\mathbb{1}(s_i=1)} \left(\frac{1}{1+e^{\eta\, a_i^{\mathrm{T}}\theta}}\right)^{\mathbb{1}(s_i=-1)}, \tag{2.4}$$

for $\theta \in \mathbb{R}^d$. We refer to γ_η as a soft tMVN distribution and generically denote it by $\mathcal{N}_{\mathcal{C}}^{s}(\mu, \Sigma)$.

In the one-dimensional case, $\gamma_\eta(\theta) = \phi(\theta|\mu,\sigma)F(\theta)$ where $\phi(x|\mu,\theta)$ is the normal density with mean μ and variance σ^2 and $F(x)$ is the logistic distribution function. This is similar to a skew normal density, except in the skew normal density, $F(x)$ is the normal distribution function instead of the logistic distribution function [Arellano-Valle and Azzalini, 2006].

It is immediate to note that γ_η is a smooth (infinitely differentiable) density supported on $\mathbb{R}^d$. Further, a simple calculation shows that

$$\nabla^2\big(-\log\gamma_\eta(\theta)\big) = \Sigma^{-1} + \sum_{i=1}^{r} \frac{\eta^2\, e^{\eta\, a_i^{\mathrm{T}}\theta}}{(1+e^{\eta\, a_i^{\mathrm{T}}\theta)^2}}\, a_i a_i^{\mathrm{T}} \succsim 0,$$

i.e., the Hessian matrix of the negative log density is positive definite. This implies that γ_η is a log-concave density, which, in particular means γ_η is unimodal. We collect these various observations about γ_η in Proposition 2.2.1.

Proposition 2.2.1. *Let γ and γ_η be respectively defined as in* (2.1) *and* (2.4)*. Then, γ_η is an infinitely differentiable, unimodal, log-concave density on $\mathbb{R}^d$. Further,*

$$\lim_{\eta\to\infty} \int_{\mathbb{R}^d} |\gamma_\eta(\theta) - \gamma(\theta)|\, d\theta = 0.$$

A proof is provided in Appendix A. The last part of Proposition 2.2.1 formalizes the intuition that γ_η approximates γ for large η by showing that the L_1 distance between γ_η and γ converges to 0 as $\eta \to \infty$. An inspection of the proof for the L_1 approximation will reveal that we haven't used any particular feature of the logistic function and the argument can be extended to other sigmoid functions.

The L_1 approximation result implies that although γ_η has a non-zero density at all points in

$\mathbb{R}^d$, the effective support is the region $\mathcal{C}$ for large values of η, and a random draw from γ_η will fall inside $\mathcal{C}$ with overwhelmingly large probability. This is because

$$\begin{aligned}\gamma_\eta(\theta \notin \mathcal{C}) &= 1 - \gamma_\eta(\theta \in \mathcal{C}) \\ &= \gamma(\theta \in \mathcal{C}) - \gamma_\eta(\theta \in \mathcal{C}) \\ &\leq \int_{\mathbb{R}^d} |\gamma_\eta(\theta) - \gamma(\theta)|\, d\theta,\end{aligned}$$

so using Proposition 2.2.1, the probability of θ falling outside of the region $\mathcal{C}$ approaches zero as η approaches infinity. To obtain a more quantitative feel for how the approximation gets better with increasing η, we set γ to be a standard bivariate normal distribution truncated to the first orthant,

$$\gamma(\theta) \propto e^{-\theta^{\mathrm{T}}\Sigma^{-1}\theta}\, \mathbb{1}_{(0,\infty)}(\theta_1)\, \mathbb{1}_{(0,\infty)}(\theta_2), \quad \Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}. \tag{2.5}$$

Figure 2.1 shows contour plots of γ (last column) along with those for γ_η for various values of η, with η increasing from left to right. Each row corresponds to a different value of ρ. It is evident that the approximation quickly improves as η increases, and stabilizes around $\eta = 100$. We later show in simulations involving substantial higher dimensions that γ_η with $\eta = 100$ continues to provide a reasonable approximation to the corresponding tMVN distribution γ.

The accurate approximation of the soft tMVN has two important consequences in our opinion. First, for any of the examples discussed in the introduction which require a sample from a tMVN within an MCMC algorithm, a sample from a tMVN can be replaced with a sample from the corresponding soft tMVN distribution; we discuss efficient strategies to sample the soft tMVN distribution in the next subsection. Second, the soft tMVN distribution can itself be used as a prior distribution for constrained parameters. As a prior, the soft tMVN replaces the hard constraints imposed by the tMVN with soft constraints, encouraging shrinkage towards the constrained region $\mathcal{C}$. Indeed, the soft tMVN distribution can be considered a global shrinkage prior [Polson and Scott, 2010] which shrinks vectors towards a pre-specified constrained region.

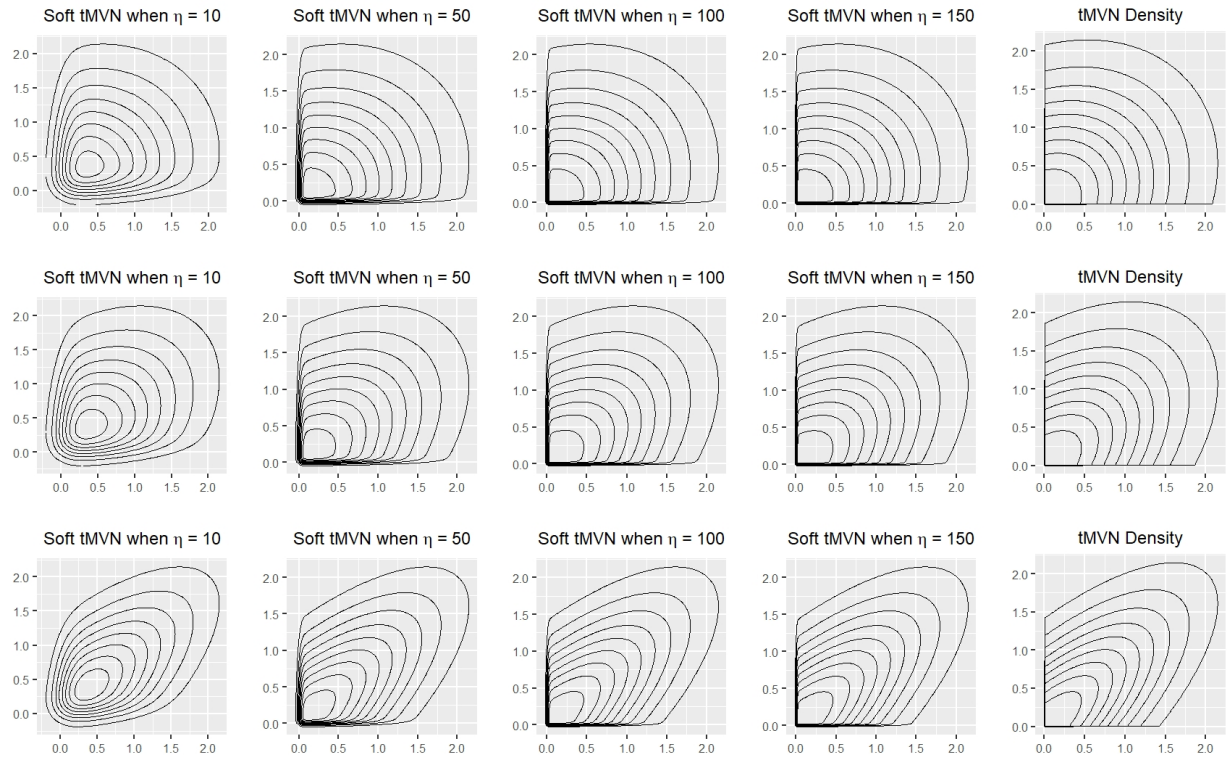

Figure 2.1: Contour plots of γ and γ_η for $\eta = 10, 50, 100$, and 150, where γ as in (2.5) is a standard bivariate normal distribution with correlation ρ, truncated to the positive orthant. The rows from top to bottom correspond to $\rho = 0.25, 0.50$, and 0.75 respectively.

The tMVN prior is conditionally conjugate for a Gaussian likelihood and the soft tMVN prior naturally inherits this conditional conjugacy. Suppose $Y \mid \theta, \sigma^2 \sim \mathcal{N}(\Phi\theta, \sigma^2 I_n)$ and $\theta \sim \mathcal{N}^s_{\mathcal{C}}(\mu, \Sigma)$ is assigned a soft tMVN prior. Then,

$$\theta \mid Y, \sigma^2, \mu, \Sigma \sim \mathcal{N}^s_{\mathcal{C}}\big((\Phi^{\mathrm{T}}\Phi/\sigma^2 + \Sigma^{-1})^{-1}\Phi^{\mathrm{T}}Y,\ (\Phi^{\mathrm{T}}\Phi/\sigma^2 + \Sigma^{-1})^{-1}\big).$$

The conditional conjugacy allows one to fit a conditionally Gaussian model with a soft tMVN prior using standard Gibbs sampling algorithms, provided one can efficiently sample from a soft tMVN distribution. We provide a detailed exposition in Section 3, with a specific application of the soft tMVN distribution as a prior in Bayesian monotone single-index models.

2.3 Sampling from the soft tMVN distribution

2.3.1 Gibbs sampler in high-dimensions

In this subsection, we propose a scalable data-augmentation blocked-Gibbs sampler to sample from a soft tMVN distribution. The proposed Gibbs sampler updates the entire θ vector in a block, unlike one-at-a-time updates for Gibbs samplers for tMVNs.

Apart from log-concavity, the other nice feature behind our choice of the logistic sigmoid function is that γ_η can be recognized as the posterior distribution of a vector of regression parameters in a logistic regression model. To see this, consider the setup of a logistic regression model with binary response $t_i \in \{0, 1\}$ and vector of predictors $W_i \in \mathbb{R}^d$ for $i = 1, \dots r$,

$$\Pr(t_i = 1 \mid \theta, W_i) = \frac{e^{W_i^{\mathrm{T}}\theta}}{1 + e^{W_i^{\mathrm{T}}\theta}}.$$

Assuming a $\mathcal{N}(\mu, \Sigma)$ prior on the vector of regression coefficients θ, the posterior distribution of $\theta \mid t, W, \mu, \Sigma$ is given by

$$e^{-\frac{1}{2}(\theta-\mu)^{\mathrm{T}}\Sigma^{-1}(\theta-\mu)} \prod_{i=1}^{r} \left(\frac{e^{W_i^{\mathrm{T}}\theta}}{1 + e^{W_i^{\mathrm{T}}\theta}}\right)^{t_i} \left(\frac{1}{1 + e^{W_i^{\mathrm{T}}\theta}}\right)^{(1-t_i)}.$$

If we now set $t_i = \mathbb{1}(s_i = 1)$ and $W_i = \eta\, a_i$, then the above density is identical to γ_η. The number of constraints r plays the role of the sample size, and the ambient dimension $d \geq r$ indicates the number of the regression parameters in this pseudo-logistic model. Thus, sampling from γ_η is equivalent to sampling from the conditional posterior of regression parameters in a high-dimensional logistic regression model, which can be conveniently carried out using the Polya–Gamma data augmentation scheme of Polson et al. [2013]. The Polya–Gamma scheme introduces r auxiliary variables $\omega_1, \dots, \omega_r$ and performs Gibbs sampling by alternatively sampling from $\omega \mid \theta, t$ and $\theta \mid \omega, t$ as follows:

1. Sample $\omega_i \mid \theta, t \sim \mathrm{PG}(1, W_i^{\mathrm{T}}\theta)$ independently for $i = 1, \dots, r$,

2. Sample $\theta \mid \omega, t \sim \mathcal{N}_d(\mu_\omega, \Sigma_\omega)$,

with

$$\Sigma_\omega = (W^{\mathrm{T}}\Omega W + \Sigma^{-1})^{-1}, \quad \mu_\omega = \Sigma_\omega(W^{\mathrm{T}}\kappa + \Sigma^{-1}\mu), \tag{2.6}$$

where $W \in \mathbb{R}^{r\times d}$ with ith row W_i^{T}, $t = (t_1, \ldots, t_r)^{\mathrm{T}}$, $\kappa = (t - 1/2)$, and $\Omega = \mathrm{diag}(\omega_1, \ldots, \omega_r)$. In 1, PG denotes a Polya–Gamma distribution which can be sampled using the `Bayeslogit` package in `R` [Polson et al., 2013]. Note that the entire θ vector is sampled in a block in step 2. The worst-case complexity of sampling from the multivariate Gaussian distribution in (2.6) is $O(d^3)$. However, exploiting the structure of μ_ω and Σ_ω, a sample from $\mathcal{N}(\mu_\omega, \Sigma_\omega)$ can be obtained with significantly less cost using a recent algorithm in Bhattacharya et al. [2016] provided $d \gg r$ and a $\mathcal{N}(0, \Sigma)$ variate can be cheaply sampled.

Define $\Phi = \Omega^{1/2}W$ and $\alpha = \Omega^{-1/2}\kappa$. Then, a sample from 2 is obtained by first sampling

$$\bar{\theta} \sim \mathcal{N}((\Phi^{\mathrm{T}}\Phi + \Sigma^{-1})^{-1}\Phi^{\mathrm{T}}\alpha,\ (\Phi^{\mathrm{T}}\Phi + \Sigma^{-1})^{-1}), \tag{2.7}$$

and setting

$$\theta = \bar{\mu} + \bar{\theta}, \quad \bar{\mu} = (\Phi^{\mathrm{T}}\Phi + \Sigma^{-1})^{-1}\Sigma^{-1}\mu. \tag{2.8}$$

First, by the Sherman–Woodbury–Morrison formula,

$$(\Phi^{\mathrm{T}}\Phi + \Sigma^{-1})^{-1} = \Sigma - \Sigma\Phi^{\mathrm{T}}(\Phi\Sigma\Phi^{\mathrm{T}} + I_r)^{-1}\Phi\Sigma.$$

Thus,

$$\bar{\mu} = \mu - \Sigma\Phi^{\mathrm{T}}(\Phi\Sigma\Phi^{\mathrm{T}} + I_r)^{-1}\Phi\mu, \tag{2.9}$$

which only requires solving a $r \times r$ system.

Sampling $\bar{\theta}$ in (2.7) can be efficiently carried out by adapting the algorithm of Bhattacharya

et al. [2016] to the present setting. The steps are:

1. Sample $u \sim \mathcal{N}(0, \Sigma)$ and $\delta \sim \mathcal{N}(0, \mathrm{I}_r)$
2. Set $v = \Phi u + \delta$
3. Solve $(\Phi\Sigma\Phi^{\mathrm{T}} + \mathrm{I}_r)w = (\alpha - v)$
4. Set $\bar{\theta} = u + \Sigma\Phi^{\mathrm{T}}w$.

It follows from Bhattacharya et al. [2016] that $\bar{\theta}$ obtained in step 4 has the desired Gaussian distribution. Barring the sampling of u in step 1, the remaining steps have a combined complexity of $O(r^2 d)$, which can be significantly smaller than d^3 when $d \gg r$. If Σ is a diagonal matrix, u can be trivially sampled with $O(d)$ cost. Even for non-diagonal Σ, it is often possible to exploit its structure to cheaply sample from $\mathcal{N}(0, \Sigma)$. For example, in the probit and multivariate probit regression context, Σ assumes the form (see subsection 2.4.2),

$$\Sigma = \begin{pmatrix} \mathrm{I}_N + HLH^{\mathrm{T}} & HL \\ LH^{\mathrm{T}} & L \end{pmatrix},$$

where L is a $q \times q$ diagonal matrix and H is an $N \times q$ (possibly dense) matrix. A sample u from $\mathcal{N}(0, \Sigma)$ is then obtained by

1. Sample $z \sim \mathcal{N}(0, \mathrm{I}_N)$ and $u_2 \sim \mathcal{N}(0, L)$ independently.
2. Set $u_1 = Hu_2 + z$ and $u = (u_1^{\mathrm{T}}, u_2^{\mathrm{T}})^{\mathrm{T}}$.

Since u is a linear transformation of (z, u_2) which is jointly Gaussian, u also has a joint Gaussian distribution. Calculating the covariance matrix of u then immediately shows that $u \sim \mathcal{N}(0, \Sigma)$. Since L is diagonal, u_2 can be sampled in $O(q)$ steps, and the matrix multiplication costs $O(Nq^2)$, so that the overall cost is $O(Nq^2)$.

2.3.2 Other strategies

In moderate dimensions, it is possible to use a Metropolis (Gaussian) random walk and its various extensions to sample from a soft tMVN distribution. In particular, given that the soft tMVN distribution can be recognized as the posterior distribution in a model with a Gaussian prior, elliptical slice sampling [Murray et al., 2010] is a viable option.

There is substantial literature on sampling from log-concave distributions using variants of the Metropolis algorithm with strong theoretical guarantees [Frieze et al., 1994, Frieze and Kannan, 1999, Lovász and Vempala, 2006a,b, Belloni and Chernozhukov, 2011]. More recently, Dalalyan [2017] and Durmus and Moulines [2019] provided non-asymptotic bounds on the rate of convergence of unadjusted Langevin Monte Carlo (LMC) algorithms for log-concave target densities. Assuming the target density is proportional to $e^{-f(\theta)}$ for some convex function f, the successive iterates of a first-order LMC algorithm takes the form

$$\theta_{k+1} = \theta_k - h\nabla f(\theta_k) + \sqrt{2h}\,\xi_{k+1}, \quad k = 0, 1, \ldots,$$

where the $\{\xi_k\}$s are independent $\mathcal{N}(0, \mathrm{I})$ variates and $h > 0$ is a step-size parameter. Clearly, $\{\theta_k\}_{k=0,1,\ldots}$ forms a discrete-time Markov chain and the results in Dalalyan [2017] and Durmus and Moulines [2019] characterize the rate at which the distribution of θ_k converges to the target density in total variation distance. Aside from the non-asymptotic bounds, another key message from their results is that the typical Metropolis adjustment as in Metropolis adjusted Langevin (MALA) [Roberts and Rosenthal, 1998] is not required for log-concave targets. Dalalyan [2017] also provides a second-order version of the LMC algorithm called LMCO which can incorporate the Hessian $\nabla^2 f$. Since both $\nabla(-\log\gamma_\eta)$ and $\nabla^2(-\log\gamma_\eta)$ are analytically tractable, it is possible to use both the LMC and LMCO algorithms to sample from γ_η.

Other than MCMC, another possible strategy to sample from γ_η is to use a multivariate generalization of the adaptive rejection sampling (ARS) [Gilks and Wild, 1992].

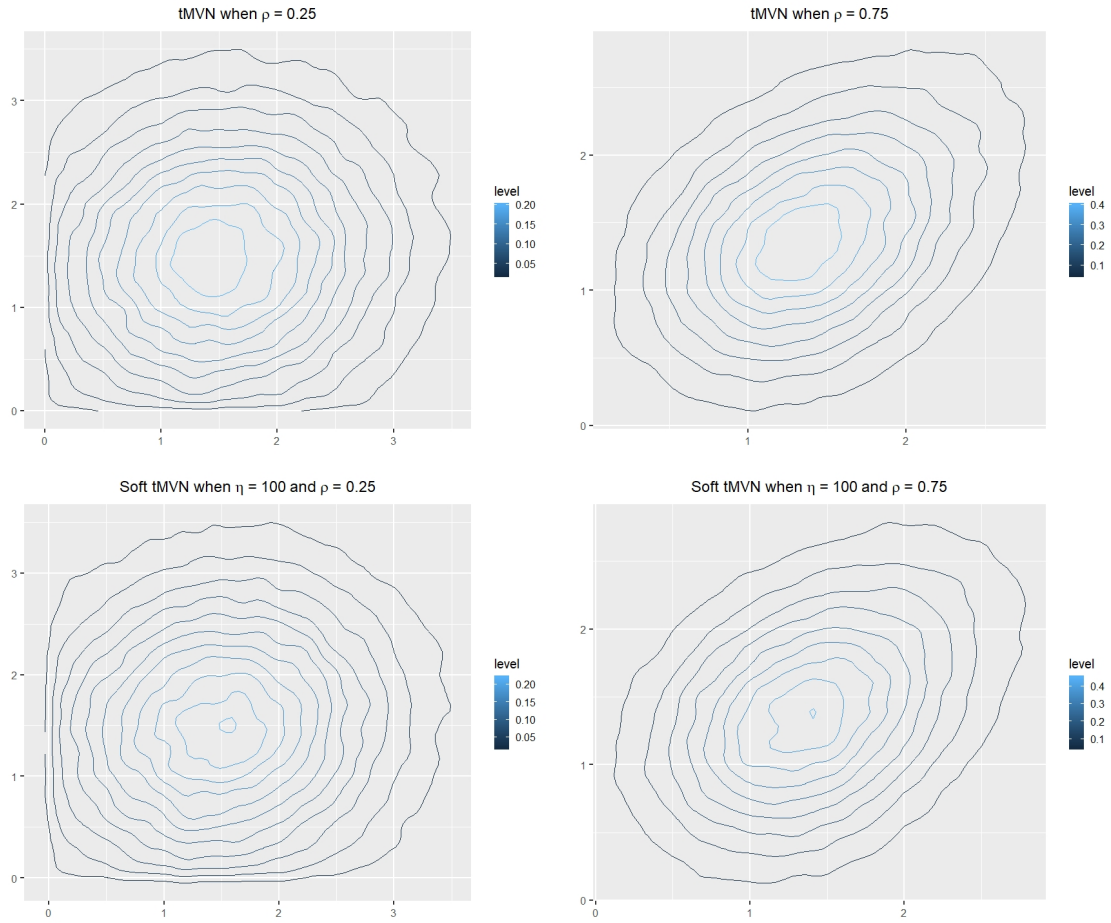

Figure 2.2: The top panel shows contour plots of a bivariate marginal of a 50-dimensional tMVN distribution with an equicorrelation covariance structure obtained using Botev's rejection sampler; the left and right figures correspond to the correlation parameter $\rho = 0.25$ and 0.75 respectively. The bottom panel shows the same for the corresponding soft tMVN distribution with $\eta = 100$, which continues to provide a good approximation.

2.4 Simulations

In this subsection, we conduct a number of simulations to empirically illustrate that the soft tMVN distribution continues to provide an accurate approximation to the tMVN distribution in high-dimensional situations. These simulations also demonstrate the scalability of the proposed Gibbs sampler.

To begin with, we first justify our continued use of $\eta = 100$ in higher dimensions. In Figure 2.1, we had provided the contour plots of a bivariate tMVN distribution and its soft tMVN

approximation with $\eta = 100$. As an obvious extension, we now consider the bivariate marginal of (θ_1, θ_2), where $\theta \in \mathbb{R}^{50}$ is drawn from a multivariate normal distribution with mean $\mu = 0$ and with a compound symmetry covariance structure, $\Sigma = (1-\rho)I_{50} + \rho 1_{50} 1_{50}^{\mathrm{T}}$, truncated to the positive orthant. We consider two choices of ρ, namely $\rho = 0.25$ and 0.75, and provide the contour plots for $\mathcal{N}_{\mathcal{C}}(\mu, \Sigma)$ and $\mathcal{N}_{\mathcal{C}}^{s}(\mu, \Sigma)$ in the top and bottom panels of Figure 2.2 respectively. The contour plots were drawn by collecting $150,000$ samples from the $\mathcal{N}_{\mathcal{C}}(\mu, \Sigma)$ and $\mathcal{N}_{\mathcal{C}}^{s}(\mu, \Sigma)$ distributions, and then retaining the first two coordinates in each case to obtain samples from the bivariate marginal. Specifically, we used the rejection sampler of Botev [2017] implemented in the `R` package `TruncatedNormal` [Botev, 2015] to draw samples from a tMVN distribution and used our data augmentation Gibbs sampler to sample from the soft tMVN distribution. The figure shows that $\eta = 100$ remains a reasonable choice in higher dimensions, and we henceforth fix $\eta = 100$ throughout. The figure also shows that the contours between the two distributions are comparable with the soft tMVN having a slightly larger peak.

Next, we provide some numerical summaries in two different settings. Due to the inherent difficulty of comparing two high-dimensional distributions, we will compare the marginal densities. Specifically, given densities f and g on $\mathbb{R}^d$ with finite mean, we consider two different measures to compare them. The first one uses the 1st Wasserstein (W_1) distance between two distributions, $W_1(f, g)$ [Villani, 2008]. The W_1 distance is defined as

$$W_1(f, g) = \inf_{(U,V) \in \mathcal{C}_{f,g}} E\|U - V\|$$

where $\mathcal{C}_{f,g}$ is the collection of all couplings between f and g, i.e., pair of random variables (U, V) with $U \sim f$ and $V \sim g$. Our first comparison metric is an average W_1 distance between the marginals,

$$D := \frac{1}{d} \sum_{i=1}^{d} \mathrm{W}_1(f_i, g_i), \tag{2.10}$$

where f_i denotes the ith marginal density of f. We used the R package `transport` to compute

the average W_1 distance between γ and γ_η, which to our convenience only requires samples from the two densities in questions. We note here that an analytic calculation is out of question since the marginal densities of both γ and γ_η lack closed-form expressions.

Our second measure is an average squared L_2 distance between the mean vectors for the two densities,

$$\xi := \frac{\|\mu_f - \mu_g\|^2}{d}, \tag{2.11}$$

with $\mu_f = \int_{\mathbb{R}^d} x f(x) dx$.

We compute D and ξ between γ and γ_η for two different covariance structures in Σ. Due to the lack of analytic expressions for the marginals for non-diagonal Σ, we resort to simulations to approximate D and ξ. The highest dimension d used in our simulations is $d = 600$; while our sampler can be scaled beyond this, the rejection sampler starts producing warning messages due to incurring small acceptance probabilities. The code for sampling from the soft tMVN distribution with both covariance structures is located at `https://github.com/aesouris/softTMVN`.

2.4.1 Probit-Gaussian Process example

For our first example, we consider $\theta \sim \mathcal{N}_n(0, \Sigma)\mathbb{1}_{\mathcal{C}}(\theta)$ where the covariance matrix Σ is formed from the Mátern kernel [Rasmussen, 2004] and $\mathcal{C} = \mathcal{C}_1 \otimes \mathcal{C}_2 \otimes \cdots \otimes \mathcal{C}_n$ where $\mathcal{C}_i$ is either $(-\infty, 0)$ or $(0, \infty)$ for $i = 1, \ldots, n$. This structure is motivated by a binary Gaussian process (GP) classification model. Suppose $Y_i \in \{0, 1\}$ is a binary response at locations s_i, modeled as $Y_i = \mathbb{1}\{Z(s_i) > 0\}$ for $i = 1, \ldots, n$, where Z is a continuous latent threshold function. In GP classification, Z is assigned a mean-zero Gaussian process prior $Z \sim GP(0, K_n)$, with $[K_n]_{ij} = K(s_i, s_j)$ and K a positive definite kernel. Here, we take K to be a Mátern kernel. Letting $Z = [Z(s_1), \ldots, Z(s_n)]^{\mathrm{T}}$, the conditional distribution of $Z \mid Y$ follows the above $\mathcal{N}_n(0, K_n)\mathbb{1}_{\mathcal{C}}(Z)$ where $\mathcal{C}_i = (-\infty, 0)$ if $Y_i = 0$ and $\mathcal{C}_i = (0, \infty)$ if $Y_i = 1$.

For the simulation, set $n = \{100, 200\}$. Let $s_i = i$ for $i = 1, \ldots, n$. We randomly sample ℓ_1 from $\{10, \ldots, n/2\}$ and ℓ_2 from $\{n/2+1, \ldots, n-10\}$ and let $Y_1, \ldots, Y_{\ell_1} = 1, Y_{\ell_1+1}, \ldots, Y_{\ell_2} = 0$,

and $Y_{\ell_2+1}, \ldots, Y_n = 1$. This is simply to mimic the situation when the true latent function Z takes positive values on $[0, a]$, negative values on $[a, b]$, and positive values again on $[b, \infty]$ for some $0 < a < b$. We set the smoothness parameter for the Mátern kernel at 3/5 and the scale parameter at 1. We then proceed to draw 5000 samples from the tMVN, $\mathcal{N}_n(0, \Sigma)\mathbb{1}_{\mathcal{C}}(\theta)$, using Botev's rejection sampler and 5000 samples from the soft tMVN, $\mathcal{N}_n^s(0, \Sigma)\mathbb{1}_{\mathcal{C}}(\theta)$, using our Gibbs sampler. The 5000 samples were collected for our method after discarding 1000 initial samples as burn-in and collecting every 100th sample to thin the chain. There is high autocorrelation in the chain, so the large thinning parameter is necessary, but this is an effecient sampler, so we are not worried about the extra sampling.

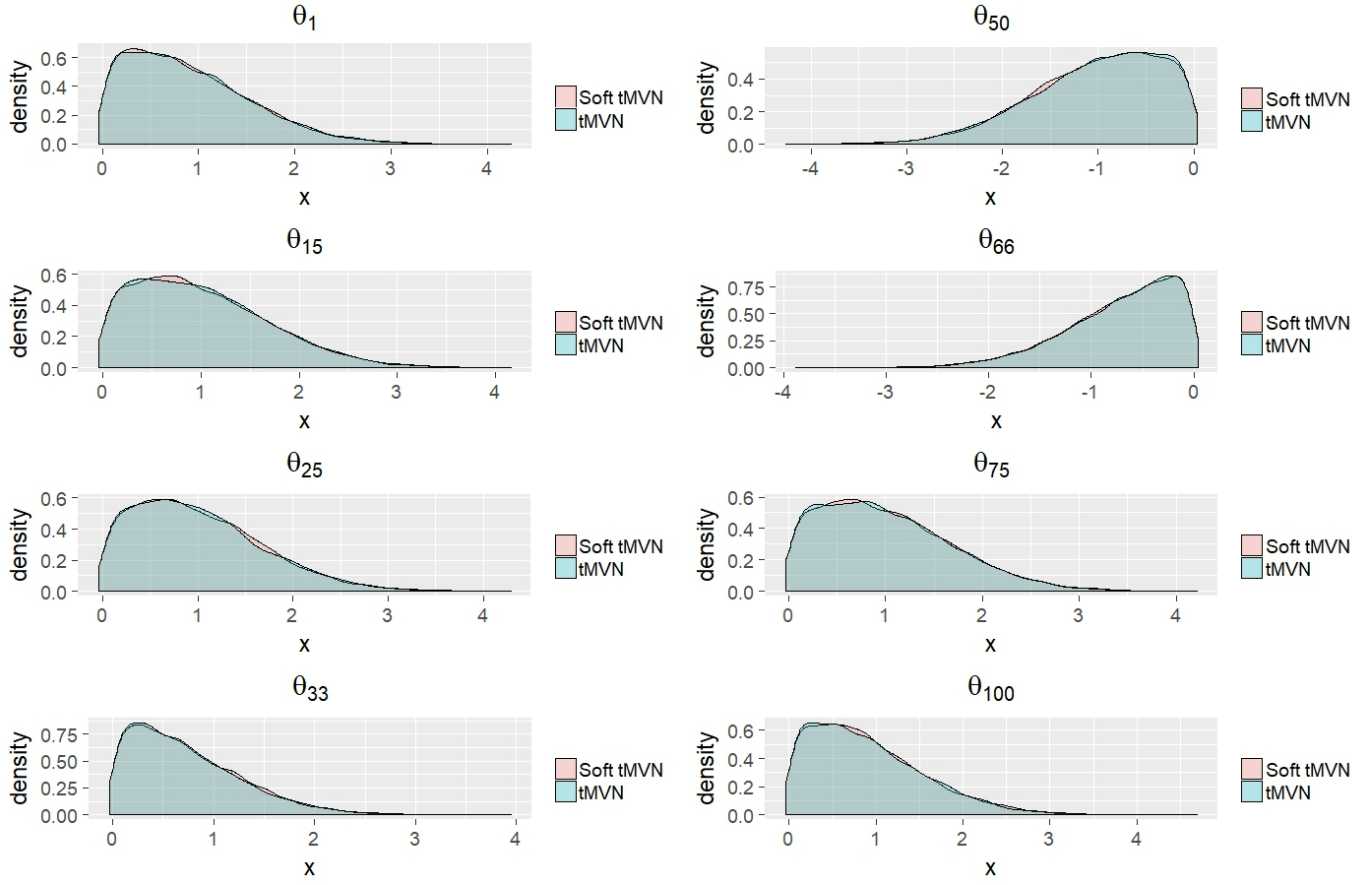

Figure 2.3: Overlapping density plot for the Probit-Gaussian Process simulation when $n = 100$. Blue denotes the tMVN distribution and pink denotes the soft tMVN distribution. The density plots are obtained using 5000 independent samples from each distribution.

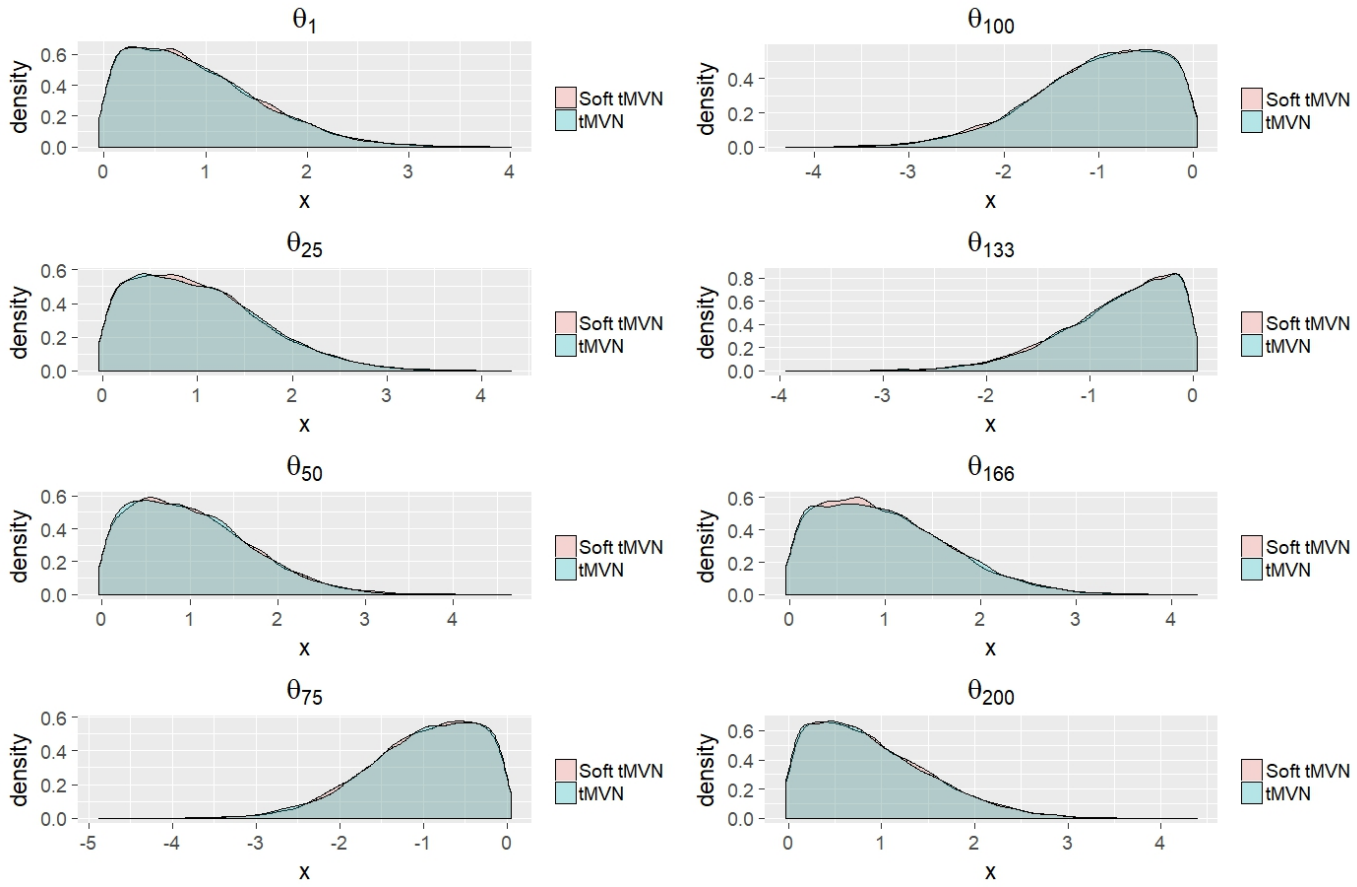

Figure 2.4: Overlapping density plot for the Probit-Gaussian Process simulation when $n = 200$. Blue denotes the tMVN distribution and pink denotes the soft tMVN distribution. The density plots are obtained using 5000 independent samples from each distribution.

Figures 2.3 and 2.4 show the marginal density plots of 8 coordinates of θ based on the 5000 samples for the two values of n respectively. The tMVN distribution is shown in blue while the soft tMVN is in pink. It is evident that for both values of n, the marginal densities are visually indistinguishable. To obtain an overall summary measure, Figure 2.5 shows the histogram of ξ, defined in equation (2.11), (left panel) and D, defined in (2.10), over 50 independent simulations. Both the histograms are tightly centered near the origin, which again suggests the closeness of the tMVN and soft tMVN distributions. As a quick comparison, the value of D between $\mathrm{N}(0, \Sigma)$ and $\mathrm{N}(0.005, \Sigma)$ for the current Σ is about 0.03 for both values of n.

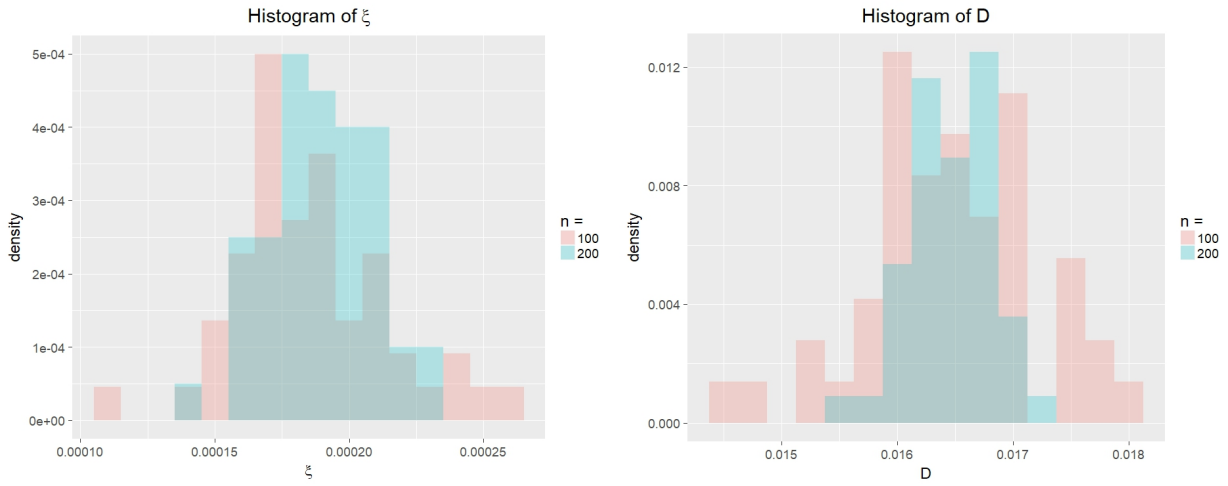

Figure 2.5: Histogram of ξ (left panel) and D (right panel) over 50 independent replicates for the Probit-Gaussian Process simulation. The pink is when $n = 100$ and the blue is when $n = 200$.

2.4.2 Probit-Gaussian example

Our second example assumes $\theta \sim \mathcal{N}_{N+P}(0, \Sigma)\mathbb{1}_{\mathcal{C}}(\theta)$ where

$$\Sigma = \begin{bmatrix} I_n + X\Lambda X^{\mathrm{T}} & X\Lambda \\ \Lambda X^{\mathrm{T}} & \Lambda \end{bmatrix},$$

$\mathcal{C} = \mathcal{C}_1 \otimes \mathcal{C}_2 \otimes \cdots \otimes \mathcal{C}_N \otimes \mathbb{R}^P$, $\mathcal{C}_i$ is either $(-\infty, 0)$ or $(0, \infty)$ for i in $1, \ldots, N$, X is an $N \times P$ matrix, and Λ is a $P \times P$ diagonal matrix.

This covariance structure is motivated by a univariate/multivariate probit model. The usual univariate probit model has binary response variables $Y_i = \{0, 1\}$ with predictors $x_i \in \mathbb{R}^d$ for $i = 1, \ldots, n$. Using the latent variable representation of Albert and Chib [1993], $Y_i = \mathbb{1}(z_i > 0)$ where z_i follows a $\mathcal{N}(x_i^{\mathrm{T}}\beta, 1)$ distribution and $\beta \in \mathbb{R}^p$. Setting a Gaussian prior on β, $\beta_j \sim \mathcal{N}(0, \lambda_j)$, the joint distribution of $\theta = [z, \beta]$ follows a Gaussian distribution. Then the conditional posterior of $\theta \mid Y, x, \lambda$ follows the above $\mathcal{N}_{N+P}(0, \Sigma)\mathbb{1}_{\mathcal{C}}(\theta)$ distribution where $X = [x_1, \ldots x_n]^{\mathrm{T}}$, $\Lambda = \mathrm{diag}\{\lambda_1, \ldots, \lambda_p\}$, $N = n$, $P = p$, and $\mathcal{C}_i = (-\infty, 0)$ if $Y_i = 0$ and $\mathcal{C}_i = (0, \infty)$ if $Y_i = 1$.

The multivariate probit model has data (y_i, x_i) where $y_i = [y_{i1}, \ldots, y_{iq}] \in \{0, 1\}^q$ is a binary

response with predictors $x_i \in \mathbb{R}^p$ for $i = 1, \dots n$. Using data augmentation, $y_{ik} = \mathbb{1}(z_{ik})$ where z_{ik} follows a $\mathcal{N}(x_i^{\mathrm{T}}\beta_k, 1)$ distribution and $\beta_k \in \mathbb{R}^p$. Assume that β_{jk} follows a $\mathcal{N}(0, \lambda_{jk})$ prior. Letting $\tilde{y}_k = [y_{1k}, \dots, y_{nk}]$, $\tilde{z}_k = [z_{1k}, \dots, z_{nk}]$, and $\lambda_k = [\lambda_{1k}, \dots, \lambda_{pk}]$, we can rewrite the model in terms of vectors instead of matrices. Let $Y = [\tilde{y}_1, \dots, \tilde{y}_q]$, $Z = [\tilde{z}_1, \dots, \tilde{z}_q]$, $\lambda = [\lambda_1, \dots, \lambda_q]$, and $\beta = [\beta_1, \dots, \beta_q]$. Then $\theta = [Z, \beta]$ follows a Gaussian distribution and the conditional distribution of θ follows the above $\mathcal{N}_{N+P}(0, \Sigma)\mathbb{1}_{\mathcal{C}}(\theta)$ where $\tilde{X} = [x_1, \dots, x_n]^{\mathrm{T}}$, $X = \text{diag}(\tilde{X})_{k=1,\dots,q}$, $\Lambda = \text{diag}(\lambda)$, $N = nq$, $P = pq$, and $\mathcal{C}_{ik} = (-\infty, 0)$ if $y_{ik} = 0$ and $\mathcal{C}_{ik} = (0, \infty)$ if $y_{ik} = 1$.

For this simulation, we sample $x_i \overset{iid}{\sim} \mathcal{N}(0, I_P)$ and $\lambda_j \sim U[1/15, 1/5]$, and then set Σ to the above form. Draw $\beta \sim \mathcal{N}(0, \Lambda)$ and $Z \sim \mathcal{N}(X\beta, I_n)$. Then if $Z_i \geq 0$, set $Y_i = 1$ and if $Z_i < 0$, set $Y_i = 0$. For both $(N, P) = \{(100, 400), (200, 400)\}$, we then proceed to draw 5000 samples from the tMVN, $\mathcal{N}_n(0, \Sigma)\mathbb{1}_{\mathcal{C}}(\theta)$, using Botev's rejection sampler and 5000 samples from the soft tMVN, $\mathcal{N}_n^s(0, \Sigma)\mathbb{1}_{\mathcal{C}}(\theta)$, using our Gibbs sampler. The 5000 samples were collected for our method after discarding 1000 initial samples as burn-in and collecting every 100th sample to thin the chain.

Figures 2.6 and 2.7 show the marginal density plots of 8 coordinates of θ based on the 5000 samples for the two combinations respectively; as before, the tMVN distribution is shown in blue while the soft tMVN is in pink. We once again see that for both combinations, the marginal densities overlap well. To obtain an overall summary measure, Figure 2.8 shows the histogram of ξ, defined in equation (2.11), (left panel) and D, defined in (2.10), over 50 independent simulations. We see that the histogram of ξ and D shifts to the right for $n = 200$ than for $n = 100$. This shift is expected as the size of the matrix X grows, and thus, the size of Σ grows. As a point of comparison, in Figure 2.9, we plot the histogram of D between $\mathcal{N}(0, \Sigma)$ and $\mathcal{N}(0.005, \Sigma)$ for the present choice of Σ and see a similar shift. We believe that the shift occurs for the probit-Gaussian motivated soft tMVN but not the probit-Gaussian process motivated soft tMVN due to structure of Σ. In the probit-Gaussian process motivated soft tMVN, Σ does not change with each trial and it has a very solid structure, while in the probit-Gaussian motivated soft tMVN, Σ changes for each trial and has a very random structure.

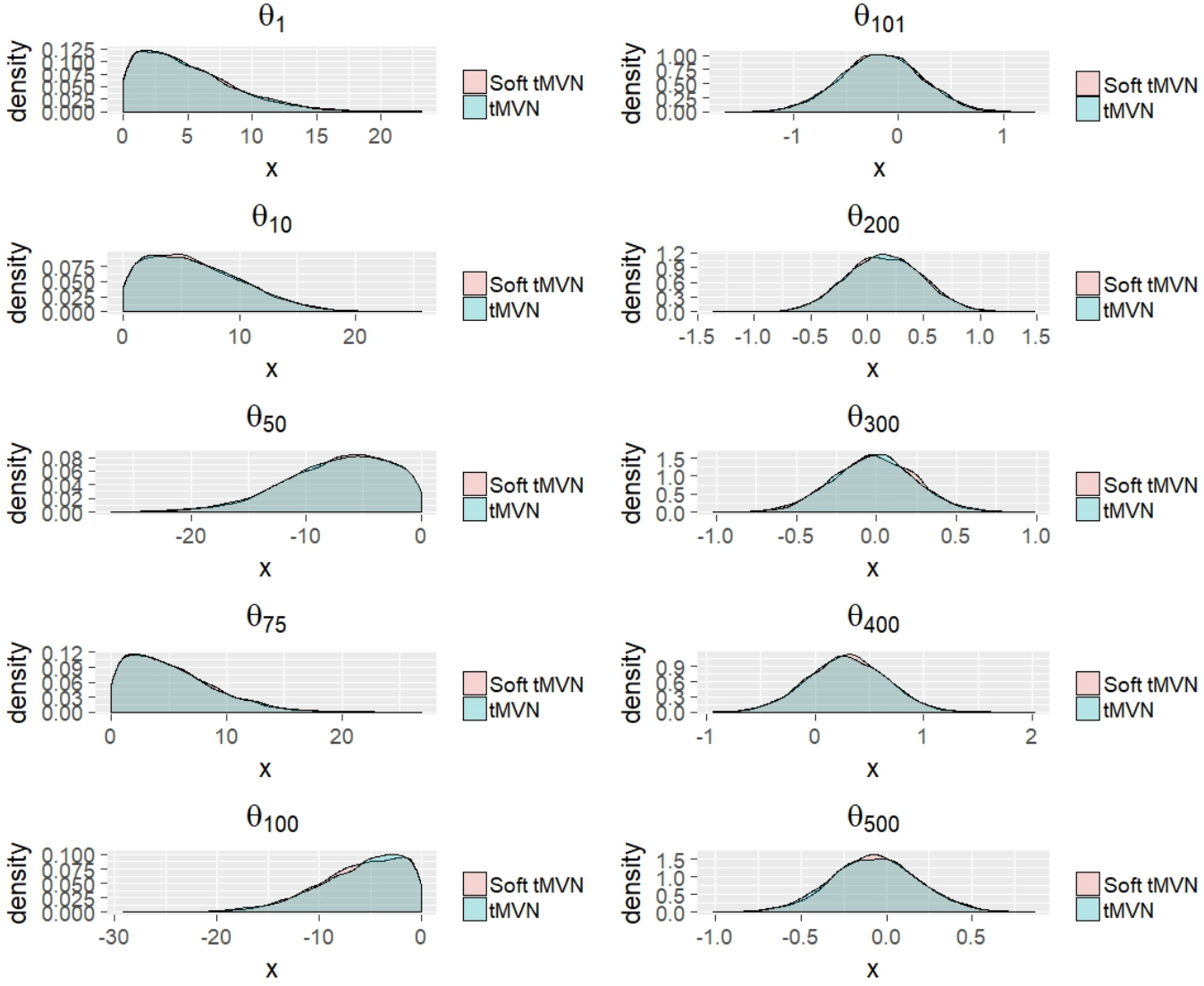

Figure 2.6: Overlapping density plot for the Probit-Gaussian simulation when $n = 100$. Blue denotes the tMVN distribution and pink denotes the soft tMVN distribution. The density plots are obtained using 5000 independent samples from each distribution.

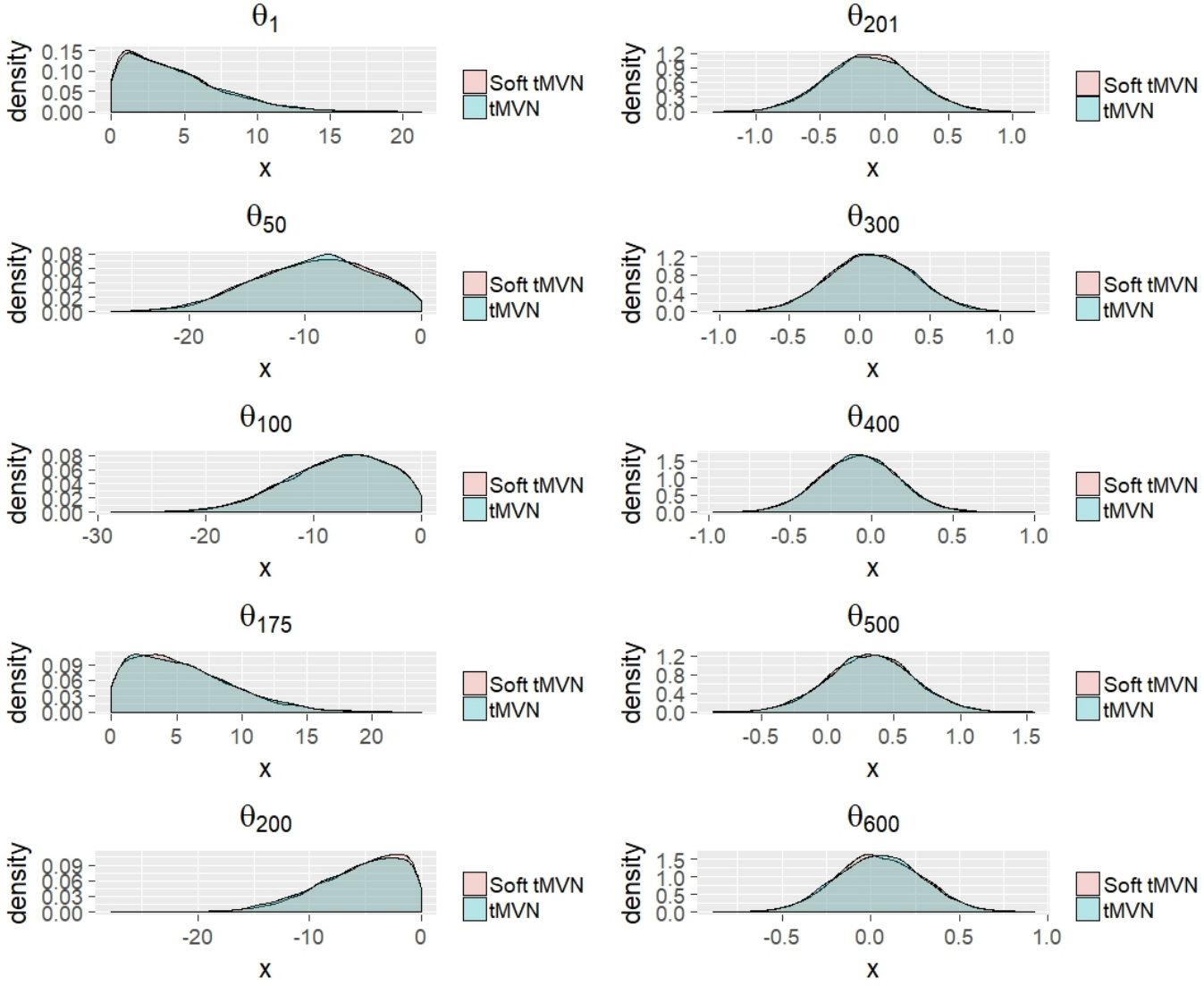

Figure 2.7: Overlapping density plot for the Probit-Gaussian simulation when $n = 200$. Blue denotes the tMVN distribution and pink denotes the soft tMVN distribution. The density plots are obtained using 5000 independent samples from each distribution.

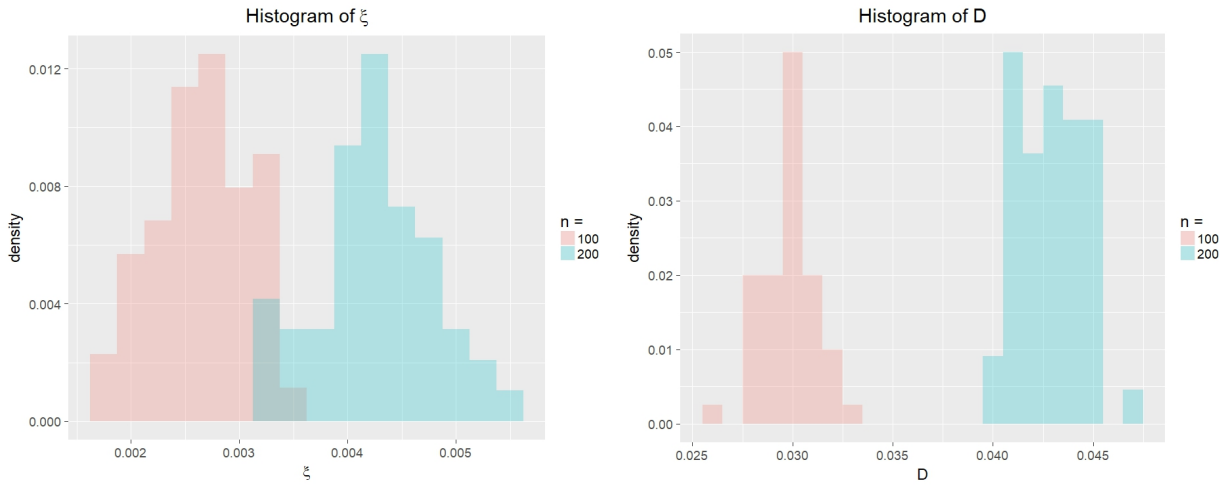

Figure 2.8: Histogram of ξ (left) and D (right) over 50 trials for the Probit-Gaussian simulation. The pink is when $n = 100$ and the blue is when $n = 200$.

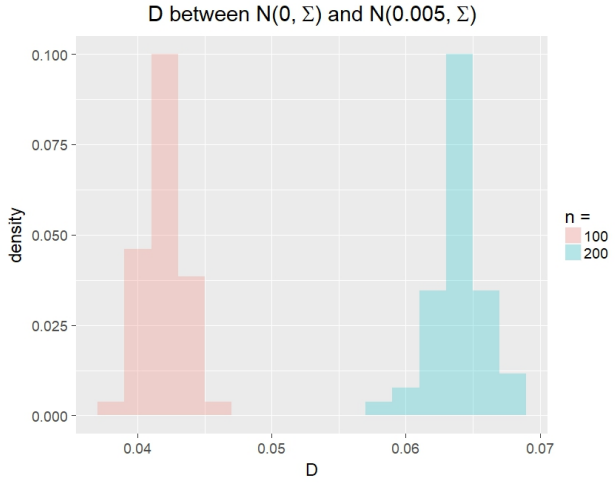

Figure 2.9: Histogram of D over 50 trials between $\mathcal{N}(0, \Sigma)$ and $\mathcal{N}(0.005, \Sigma)$ where Σ is the same as in the Probit-Gaussian simulations. The pink is when $n = 100$ and the blue is when $n = 200$. This is used for comparison with Figure 2.8.

2.5 Discussion

In this section, we have presented the soft tMVN distribution, which provides a smooth approximation to the tMVN distribution with linear constraints. Our theoretical and empirical results suggest that the soft tMVN distribution offers a good approximation to the tMVN distribution in high dimensional situations.

3. APPLICATIONS TO BAYESIAN CONSTRAINED ESTIMATION

3.1 Introduction

The truncated multivariate normal (tMVN) distribution is routinely used as a prior distribution on model parameters in Bayesian shape-constrained regression. Structural constraints, such as monotonicity and/or convexity, are commonly induced by expanding the function in an appropriate basis where the constraints can be induced by imposing *linear constraints* on the coefficients; some examples of such a basis include piecewise linear functions [Dunson and Neelon, 2003], splines [Cai and Dunson, 2007], Bernstein polynomials [Wang and Ghosh, 2012], and compactly supported basis functions [Maatouk and Bay, 2017, Zhou et al., 2019a]. Under a Gaussian or scale-mixture of Gaussian error distribution, the conditional posterior of the basis coefficients once again turns out to be truncated normal with linear constraints, necessitating sampling from a tMVN distribution for posterior inference.

Since the soft tMVN distribution developed in Section 2 is also conditionally conjugate to a Gaussian likelihood, one may use it as a prior distribution on the basis coefficients in Bayesian shape-constrained regression problems. In this section, we develop an approximate MCMC [Johndrow et al., 2015] algorithm which replaces the tMVN distribution with the soft tMVN distribution as the prior in a Bayesian shape-constrained regression problem. For illustration purpose, we consider a monotone single-index model considering its usefulness in practical applications, noting that the methodology can be extended to more standard constrained regression applications such as estimation of bounded, monotone, or convex/concave functions. We pick the monotone single-index model example due to limited previous treatment from a Bayesian perspective. Moreover, this example nicely brings out the computational advantages of using a soft tMVN prior.

The rest of the section is as follows. In subsection 3.2 we introduce the monotone single-index model. In subsection 3.3 we describe the prior specification. Subsection 3.4 contains a simulation that exemplifies the usefulness of the soft tMVN distribution as an approximation to the tMVN

distribution. We conclude in subsection 3.5 with a discussion.

3.2 Monotone single-index model

Given response-covariate pairs $\{(y_i, x_i)\}_{i=1}^n \in \mathbb{R} \times \mathbb{R}^p$, a Gaussian single-index model [Antoniadis et al., 2004, Chen and Samworth, 2016, Gramacy and Lian, 2012, Wang, 2009, Yu and Ruppert, 2002] assumes the form

$$y_i = f(x_i^{\mathrm{T}}\alpha) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2), \quad i = 1, \ldots, n, \tag{3.1}$$

where $f : \mathbb{R} \to \mathbb{R}$ is an unknown link function and $\alpha \in \mathbb{R}^p$ an unknown coefficient vector. Throughout, we assume the covariates to be standardized. The single-index model provides a bridge between linear and non-linear modeling by first linearly projecting the high-dimensional vector of predictors to the real line and then modeling the response as a non-linear function of the projection. The model (3.1) is clearly non-identifiable without further restrictions; we follow a standard prescription to impose a unit norm restriction, $\|\alpha\| = 1$, on α.

We consider a monotone single-index model [Cavanagh and Sherman, 1998, Ahn et al., 1996, Balabdaoui et al., 2019, Foster et al., 2013, Luo and Ghosal, 2016] where the link function f is monotone non-decreasing. Monotone single-index models have widespread applications in biomedical science, e.g. find gene-gene interactions [Luss et al., 2012] and to study the relationship between risk factors for survival with leukemia [Schell and Singh, 1997]. To model f, we use a Bernstein polynomial basis noting that other basis functions mentioned in the introduction can also be used. Using the Bernstein polynomial basis, there are established sufficient conditions which enforce f to be monotonic. Define, for $j = 0, \ldots, M$,

$$B_{M,j}(u) = \binom{M}{j} u^j (1-u)^{M-j}, \quad u \in [0, 1],$$

so that the Bernstein polynomial of degree M is

$$B_M(u) = \sum_{j=0}^{M} \theta_j B_{M,j}(u).$$

If

$$\theta_0 \leq \theta_1 \leq \cdots \leq \theta_M, \tag{3.2}$$

then $B_M(u)$ is non-decreasing [Chak et al., 2005].

To apply the Bernstein polynomial basis to our setting, we need some preprocessing as described below. Since $|x_i^{\mathrm{T}}\alpha| \leq \|x_i\| \, \|\alpha\| = \|x_i\|$ by the Cauchy-Schwarz inequality and the identifiability restriction respectively, if we let $c = \max_i \|x_i\|$ and transform $\tilde{x_i} = x_i/c$, we have $|\tilde{x_i}^{\mathrm{T}}\alpha| \leq 1$. Hence, we need to perform a change of variable to transform the support of the Bernstein polynomial to $[-1, 1]$. To that end, we write $B_{M,j}(u) = p_j(u)/(M+1)$ for $u \in [0, 1]$, where $p_j(u)$ is the density of a Beta$(j+1, M-j+1)$ distribution. Letting $T = 2U - 1$ for $U \sim$ Beta$(j+1, M-j+1)$, the density of T is $q_j(t) = \frac{1}{2} p_j\{(t+1)/2\}$ for $t \in [-1, 1]$. Let $\tilde{B}_{M,j}(t) = q_j(t)/(M+1)$ for $j = 0, \ldots, M$ represent the transformed Bernstein polynomial basis and define our monotone single-index model as

$$y_i = \tilde{B}_M(\tilde{x}_i^{\mathrm{T}}\alpha) + \epsilon_i, \quad \tilde{B}_M(t) = \sum_{j=0}^{M} \theta_j \tilde{B}_{M,j}(t), \quad t \in [-1, 1]. \tag{3.3}$$

Under the order-restriction on the basis coefficients in (3.2), $\tilde{B}_M(\cdot)$ remains non-decreasing. Set $\psi_0 = \theta_0$, $\psi_1 = \theta_1 - \theta_0, \ldots, \psi_M = \theta_M - \theta_{M-1}$, so that (3.2) is equivalent to $\psi_k \geq 0$ for $k = 1, \ldots, M$. Thus the non-decreasing constraint can be written in terms of $\psi = [\psi_0, \ldots, \psi_M]^{\mathrm{T}}$. Let A be an $(M+1) \times (M+1)$ lower triangular matrix where all the lower triangle elements and diagonal elements are 1. Then $A\psi = \theta$ where $\theta = [\theta_0, \ldots, \theta_M]^{\mathrm{T}}$.

To place the monotone single-index model (3.3) in vectorized notation, let $\tilde{B}_M^i = [\tilde{B}_{M,0}(\tilde{x}_i^{\mathrm{T}}\alpha), \ldots, \tilde{B}_{M,M}(\tilde{x}_i^{\mathrm{T}}\alpha)]^{\mathrm{T}}$ and $\mathbb{B}_\alpha = [\tilde{B}_M^1, \ldots \tilde{B}_M^n]^{\mathrm{T}}$ so that $\mathbb{B}_\alpha$ is a $n \times (M+1)$ matrix, with the subscript serving as a reminder that $\mathbb{B}_\alpha$ depends on α. Then letting $Y = [y_1, \ldots, y_n]^{\mathrm{T}}$,

(3.3) can be equivalently represented as

$$Y = \mathbb{B}_\alpha \theta + \epsilon = \mathbb{B}_\alpha A \psi + \epsilon.$$

3.3 Prior specification

Our prior specification on the model parameters (ψ, α, σ^2) assumes the form $\pi(\psi, \alpha, \sigma^2) = \pi(\psi)\,\pi(\alpha)\,\pi(\sigma^2)$. We consider two different priors on ψ: (i) a tMVN prior $\mathcal{N}_{\mathcal{C}}(0, 25I_{M+1})$, and (ii) a soft tMVN prior $\mathcal{N}^s_{\mathcal{C}}(0, 25I_{M+1})$, where in both cases $\mathcal{C} = \mathbb{R} \otimes [0, \infty)^M$. Next, we set $\alpha = \beta / \|\beta\|$ and assign a standard Gaussian prior on β. Finally, we consider a inverse-Gamma prior on σ^2 with mean 1 and variance 10. For sake of future reference, we refer to the joint prior on (ψ, α, σ^2) corresponding to cases (i) and (ii) by π^h and π^s respectively, with the superscripts indicative of a usual (hard) or soft tMVN prior on the constrained parameter.

We employ a Metropolis-within-Gibbs algorithm to sample from the posterior distribution with either prior. For π^h, the conditional posterior $\psi \mid \sigma^2, \alpha$ is $\mathcal{N}_{\mathcal{C}}(\mu_\psi, \Sigma_\psi)$, while the same for π^s is $\mathcal{N}^s_{\mathcal{C}}(\mu_\psi, \Sigma_\psi)$, where

$$\Sigma_\psi = \left(\frac{1}{\sigma^2} D_\alpha^{\mathrm{T}} D_\alpha + \frac{1}{25} I_{M+1} \right)^{-1}, \quad \mu_\psi = \frac{1}{\sigma^2} \Sigma_\psi D_\alpha^{\mathrm{T}} Y, \quad D_\alpha = \mathbb{B}_\alpha A.$$

The conditional distribution of $\sigma^2 | \psi, \alpha$ is inverse-Gamma in both cases. To sample from $\alpha | \sigma^2, \psi$, we use a Metropolis step with the proposal density on β as $J(\beta^t | \beta^{t-1}) \sim \mathcal{N}(\beta_{t-1}, 0.01^2 I)$. The proposal standard deviation of 0.01 was chosen to give an acceptance probability around 0.35 for β.

3.4 Simulation

The following simulation compares the Metropolis-within-Gibbs algorithms for the priors π^h and π^s respectively. We generate data from the model (3.3) with $n = 800$, $p = 5$, $M = 20$, and a set of true parameter values $\psi_0, \alpha_0, \sigma_0^2$. We set $\sigma_0 = 0.1$ and $\alpha_0 = \beta_0 / \|\beta_0\|$ with β_0 drawn from a standard Gaussian distribution. Finally, we set $\theta_0 \in \mathbb{R}^{21}$ equal to the vector where the first six

entries are -1, then -0.5, then the next seven entries are 0, then 0.5, then the last six entries are 1. We consider 30 independent replicates for model fitting and perform out-of-sample prediction on a single separate dataset of size 200.

We set $\eta = 500$ for the soft tMVN prior π_s. We observed sensitivity for smaller values of η in this context; something that we didn't encounter earlier, possibly due to the more difficult sampling problem involved here[1]. For each of the 30 replicates, we run the Gibbs samplers for π_h and π_s outlined above to collect 1000 posterior samples each. These 1000 samples are after a burn-in period of 1000 and after thinning the chain by 100. The 1000 samples are used to calculate the posterior mean of α, $\hat{\alpha}$, and the posterior mean of θ, $\hat{\theta}$. For π_h, we use the rejection sampler of Botev [2017] implemented in the `R` package `TruncatedNormal` [Botev, 2015] to draw samples from the tMVN distribution, while for π_s, we use the data augmentation Gibbs sampler defined in subsection 2.3 to sample from the soft tMVN distribution. The code to run both Gibbs samplers can be found at `https://github.com/aesouris/softTMVN`.

In terms of statistical performance, the two samplers were comparable. The average out-of-sample prediction error for the soft tMVN prior across the 30 replicates was 0.005 with a standard deviation of 0.0106, while the same numbers for the tMVN prior were 0.002 and 0.0066 respectively.

	α-ESS	ψ-ESS	run-time (in hours)
soft tMVN prior	253.6625	686.0742	$3.78_{0.0026}$
tMVN prior	168.4741	796.6799	$15.45_{3.8204}$

Table 3.1: The first two columns report the average effective sample sizes (out of 1000 MCMC samples) for α and ψ for the two Gibbs samplers. The average is over both the parameter entries as well as the 30 replicates. The final column reports the run-time (in hours) for the respective Gibbs samplers to collect 1000 posterior samples, with the subscript denoting the standard deviation across replicates.

Table 3.1 reports the effective sample sizes for α and ψ as well as the run-time for the two

[1] See the Appendix B for an example with a smaller value of η.

Gibbs samplers. The two samplers are similar in terms of the effective sample sizes; however the Gibbs sampler for the tMVN prior has almost 5 times the run-time of the soft tMVN sampler. The mixing is slow for either samplers which is indicative of a general issue for problems with constrained parameter spaces; remember the 1000 posterior samples are collected with a thinning size of 100. Although a formal proof is beyond the scope of this dissertation, empirical evidence suggests that the constrained parameters inside the Gibbs sampler may get stuck into regions of low probability, and it can take a long time to escape these regions. Specifically, we see that Botev's state-of-the-art rejection sampler can sometimes take exceedingly long to make a single move; note the variability in the run-time across the 30 trials in Table 3.1. While our chain also suffers from a similar slow mixing, it has substantially better per-iteration cost which makes it possible to run it for a large path-length to collect a substantial number of effective samples. The computational advantage becomes even more pronounced for higher dimensions; we do not report a simulation with a higher dimension M since the tMVN sampler takes exceedingly long to run.

3.5 Discussion

In this section, we have illustrated that the soft tMVN distribution is a more computationally viable alternative prior to the usual tMVN prior in a monotone single-index model, especially in complex problems where an MCMC algorithm may get stuck in regions of very low probability, making it difficult to move. The monotone single-index model example illustrates this phenomenon and we expect it to be more widely prevalent in Bayesian constrained problems.

4. BAYESIAN SOFTMAX-AFFINE CONVEX REGRESSION

4.1 Introduction

Convex regression is useful in many applications, including economics, survival analysis, statistics, and optimization. In economics, production functions [Skiba, 1978, Varian, 1984], consumer preferences [Meyer and Pratt, 1968], and utility functions [Matzkin et al., 1991] are concave functions, which are the negative of a convex function. In survival analysis, the hazard rate and failure rate are convex functions [Forbes et al., 2011]. In statistics, density function are log-concave [Cule et al., 2010]. In optimization, geometric programming requires convex approximations [Boyd and Vandenberghe, 2004].

Traditionally, there have been fewer methods for multivariate convex regression compared to nonparametric regression methods. The earliest method is the least squares estimator (LSE; Hildreth [1954], Holloway [1979]). Another approach is to place a positive semi-definite restriction on the Hessian of the estimator [Henderson and Parmeter, 2009, Roy et al., 2007, Aguilera and Morin, 2009]. Hannah and Dunson [2013] proposed the Convex Adaptive Partitioning (CAP) and fast CAP methods. The CAP method adaptively partitions the covariate space, fits hyperplanes within each partition, then estimates the convex function by taking the maximum of the hyperplanes. Mazumder et al. [2019] uses an algorithmic framework based on the augmented Lagrangian method to solve the LSE in a scalable way.

In the Bayesian literature, convex regression is a specific example of shape-constrained regression where shape-constrained regression imposes a structural constraint such as monotonicity or convexity on the regression function. These structural constraints are usually performed by expanding a set of basis functions and then placing a constraint on their coefficients. Some examples of basis functions used for convex regression include Bernstein polynomials [Chak et al., 2005, Wang and Ghosh, 2012], compactly supported basis functions [Maatouk and Bay, 2017, Zhou et al., 2019b], regression splines [Meyer et al., 2011], and restricted splines [Shively et al., 2011].

In contrast to the basis function approaches above, Hannah and Dunson [2011] proposed a Bayesian version of their CAP method, called Multivariate Bayesian Convex Regression (MBCR), where they model the unknown convex function as the maxima of hyperplanes. They placed a prior on the number of components, as well as the slopes and intercepts for each of the individual hyperplanes. They proposed a reversible jump MCMC (RJMCMC) algorithm to sample from the posterior distribution.

We introduce the softmax-affine convex (SMA) regression which approximates a convex function using the softmax function. The softmax function is a smooth function that approximates the maximum function. The SMA regression method is similar to CAP and MBCR, but we replace the maximum function with the softmax function. One of the greatest advantages of this replacement is that the softmax function is smooth, so gradients can be computed. Then, default Hamiltonian Monte Carlo (HMC) methods can be used. The SMA regression is a fully Bayesian model, so we are able to get both posterior predictions and credible intervals for the function. It also works well for large sample sizes.

In subsection 4.2, we introduce the softmax function and SMA regression. Subsection 4.3 contains the methods utilized for checking convergence. In subsection 4.4, we run two different simulations. One compares our method to other convex regression methods like LSE (Hildreth [1954], Holloway [1979]) and CAP and Fast CAP [Hannah and Dunson, 2013]. The other simulation illustrates how to choose hyperparameters. Finally, subsection 4.5 concludes the section with a discussion.

4.2 SMA Regression

Consider a regression model

$$y_i = g(x_i) + \epsilon_i, \quad i = 1, \ldots, n, \tag{4.1}$$

where the response $y_i \in \mathbb{R}$ and the predictor $x_i \in \mathcal{X} \subset \mathbb{R}^d$. We shall assume the errors ϵ_i to be independent and normally distributed with a common variance σ^2 throughout this section, although

extensions to the heteroskedastic case is straightforward. We are interested in the situation where g is a convex function, i.e., for any $x, x' \in \mathcal{X}$ and $t \in (0, 1)$,

$$g\big(tx + (1-t)x'\big) \leq tg(x) + (1-t)g(x').$$

Hannah and Dunson [2013] proposed a convex adaptive partitioning (CAP) approach for such multivariate convex regression. The basic idea behind CAP can be described as follows. If a convex function g is differentiable, then for all $x, x' \in \mathcal{X}$, $g(x') \geq g(x) + (x' - x)^{\mathrm{T}} \nabla g(x)$, i.e., the curve always lies above its tangent line at any point x. Depending on the curvature of g, the linear approximation at x given by $g(x) + (\cdot - x)^{\mathrm{T}} \nabla g(x)$ will provide an accurate approximation in a neighborhood of x and will deteriorate beyond that. This is where CAP uses a second property of convex functions that the maxima of a collection of convex functions is again convex [Boyd and Vandenberghe, 2004]. CAP then proceeds to adaptively partition the input space. Within each partition, it fits a hyperplane, and the overall function is estimated as a maxima of the fitted hyperplanes. We note here that the development and theoretical guarantees for CAP do not require differentiability and can instead work with the weaker notion of sub-differentiability. Hannah and Dunson [2013] also proposed a Fast CAP approach which performs early stopping and is thus computationally more efficient. CAP and fast CAP appear to show significant improvements over other multivariate convex regression methods (see table 4.1) and are able to scale well to large data [Hannah and Dunson, 2013].

CAP can be also placed inside a model based formulation [Hannah and Dunson, 2011] as follows. Consider modeling g as

$$g(x) = \max_{k \in 1,\ldots,K} \big(a_k + b_k^{\mathrm{T}} x\big), \quad x \in \mathcal{X}. \tag{4.2}$$

Clearly, g as defined above is convex. The number of hyperplanes K and the hyperplane specific intercepts and slopes $\{(a_k, b_k)\}_{k=1}^{K}$ are the model parameters. Hannah and Dunson [2011] conducted a Bayesian inference in this setup which they called Multivariate Bayesian Convex Re-

gression (MBCR). Since the number of model parameters varies with K, they used a reversible jump MCMC algorithm [Green, 1995] to sample from the posterior distribution. Bhattacharya et al. [2019] showed that MBCR achieves the minimax rate of estimation for a large class of convex functions, even under mild model misspecification.

In this section, we propose a simple modification to MBCR that enables us to use convenient off-the-shelf Hamiltonian MCMC samplers. The main idea is to replace the max function with a softmax function. The softmax function $F_\beta : \mathbb{R}^K \to \mathbb{R}$ is defined as

$$F_\beta(u_1, \ldots, u_K) = \frac{1}{\beta} \log \left(\sum_{k=1}^{K} e^{u_k^{\mathrm{T}} \beta} \right), \quad u \in \mathbb{R}^K.$$

The parameter $\beta > 0$ can be thought of as an inverse-temperature parameter. For any fixed $u = (u_1, \ldots, u_K)^{\mathrm{T}}$, $F_\beta(u_1, \ldots u_K)$ approaches $\max\{u_1, \ldots u_K\}$ as β approaches ∞. In fact, we have the following global two-sided bound

$$\max\{u_1, \ldots u_K\} \leq F_\beta(u_1, \ldots, u_K) \leq \max\{u_1, \ldots u_K\} + \frac{\log K}{\beta}.$$

Thus, for large β, the softmax function F_β serves as an accurate proxy of the max function. Observe also that the softmax function is smooth, unlike the max function. This is important for gradient computations later on.

We propose to use the softmax approximation in the model (4.2). Specifically, define

$$g_\beta(x \mid \theta) = F_\beta(a_1 + b_1^{\mathrm{T}} x, \ldots, a_K + b_K^{\mathrm{T}} x),$$

where $\theta = \{\theta_k\}_{k=1}^K$ with $\theta_k = (a_k, b_k^{\mathrm{T}})$ are the unknown model parameters. Since composition with an affine function preserves convexity, $g_\beta(\cdot \mid \theta)$ is a convex function. Moreover, $g_\beta(\cdot \mid \theta)$ is smooth as a function of θ, which in particular implies the log-likelihood function can be differentiated with respect to θ. This enables us to perform posterior computation using `rstan` [Team et al., 2016], which implements a default Hamiltonian Monte Carlo (HMC; Duane et al. [1987]) algorithm.

HMC is a type of MCMC method for sampling from a posterior distribution based on the motion of molecules using Hamiltonian dynamics from physics. The Hamiltonian equations depend on the gradient of the log posterior, so having an exact expression for the gradient is important [Neal et al., 2011].

We specify our prior distributions as

$$\sigma^2 \sim IG(2.1, 1.1)$$
$$\theta_k \sim N(0, \tau_k^2 I_{p+1})$$
$$\tau_k^2 \sim IG(2.1, 1.1)$$

where IG is the inverse gamma prior, I_{p+1} is the identity matrix with dimension $p+1$, τ_k^2 are independent with mean 1 and variance 10, and the θ_k are independent. The number of components K must be specified. We also tried using a Cauchy prior on both σ^2 and τ_k^2, but it did not appear to improve the posterior estimates and the posterior sampler took longer to run. Thus, we decided to stick to the conjugate inverse gamma prior. Unless otherwise specified, we fix $\beta = 10$ and choose K using predictive cross validation. See subsection 4.4.2 for more details.

4.3 Checking Convergence

In order to help check the convergence of the HMC sampler, we have Stan automatically compute $g(\tilde{\theta}|X)$ and $g(\tilde{\theta}|X_{test})$ where $g(\theta|X)$ is the posterior distribution, $\tilde{\theta}$ are posterior samples of θ given everything else, and X_{test} is a matrix of new X values that is passed to Stan but is not used in the HMC sampler to estimate θ. Denote $\hat{y}$ as the posterior mean of $g(\tilde{\theta}|X)$ and $\hat{y}_{test}$ as the posterior mean of $g(\tilde{\theta}|X_{test})$.

Since θ has independent normal priors, there are identifiability issues with θ. However, we are interested in estimating $g(\theta|X)$ (and hence y) and not θ, so we are not concerned with this identifiability issue. When checking the convergence of the HMC sampler, we cannot check the convergence of θ, instead, we must check the convergence of $g(\theta|X)$.

To help illustrate this, we run the HMC sampler on a small dataset and present some conver-

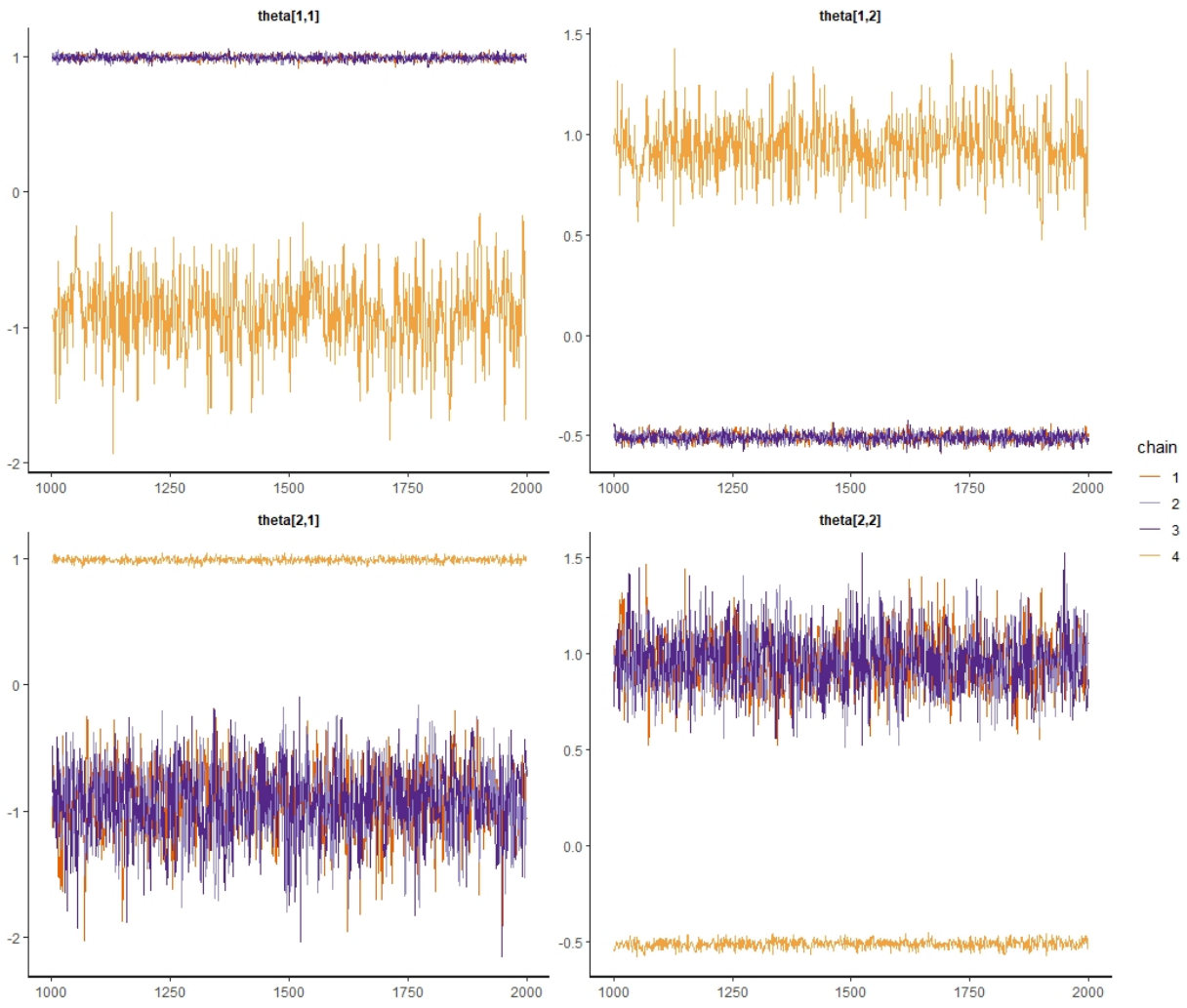

Figure 4.1: The traceplots of θ for 4 chains each with 1000 samples after 1000 burnin samples.

gence diagnostics. We set

$$\theta = \begin{bmatrix} 1 & -0.5 \\ -1 & 1 \end{bmatrix}, \tag{4.3}$$

$\sigma_y^2 = 0.5^2$, and draw $x_i \sim N(0,1)$ for $i = 1, \ldots, 800$ with an appended column of 1's at the front. We set X_{test} as a sequence from -4 to 4 by increments of 0.05, and tell the Stan model that $K = 2$, the true number of components. Let the HMC sampler run 4 chains, each with 1000 burnin samples and 1000 after burnin samples.

The traceplots for θ are contained in Figure 4.1 and show a clear label switching problem, but

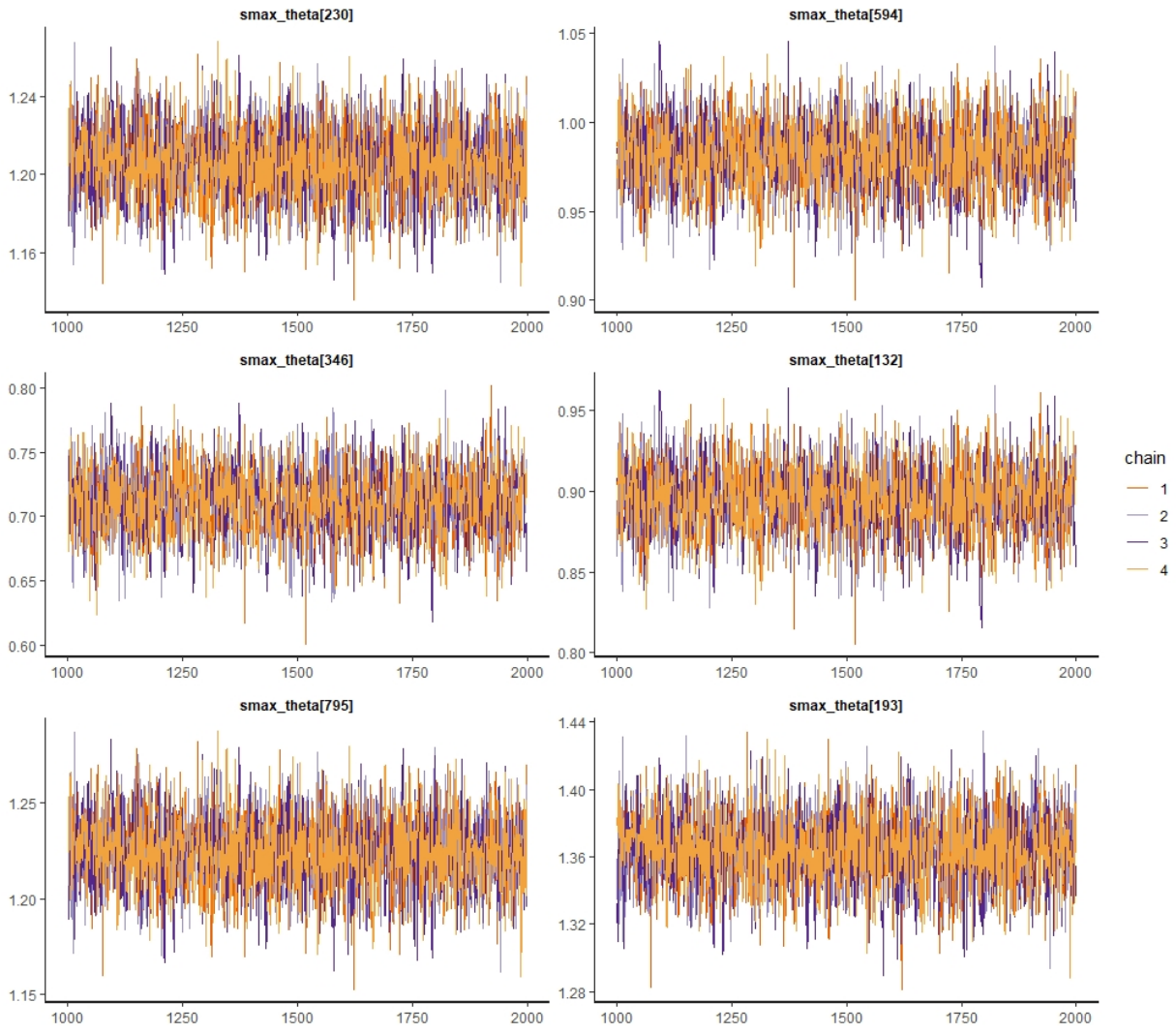

Figure 4.2: The traceplots of $g(\tilde{\theta}|x_i)$, for 6 random values of i and for 4 chains each with 1000 samples after 1000 burnin samples.

otherwise the chains are converging well. The estimated effective sample size was 2 for each theta and the potential scale reduction factor, $\hat{R}$, was greater than 3.5 for each theta where an $\hat{R}$ value greater than 1.1 indicates convergence problems. This is expected when there is a label switching problem. Instead, to check the convergence of the HMC sampler, we must check the convergence of $g(\theta|X)$.

Figure 4.2 contains the traceplots for $g(\tilde{\theta}|x_i)$ for 6 random values of i. The traceplots appear

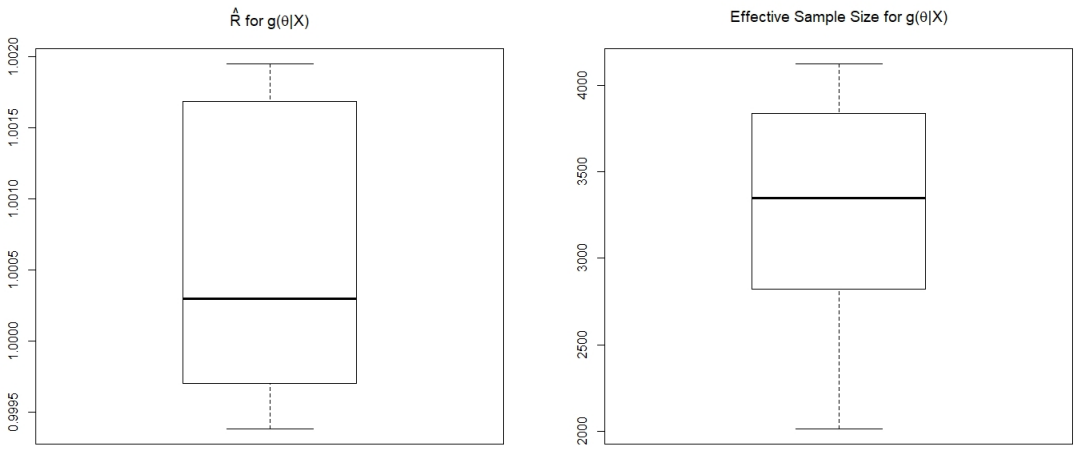

Figure 4.3: Boxplots of the values of $\hat{R}$ (left) and effective sample size (right) of $g(\tilde{\theta}|X)$ for 4 chains each with 1000 samples after 1000 burnin samples.

to converge. Figure 4.3 contains boxplots of the $\hat{R}$'s and effective sample sizes for the 800 values of $g(\tilde{\theta}|X)$. $\hat{R}$ is essentially 1 for each $g(\tilde{\theta}|x_i)$ and the effective sample size is greater than 2000 for $g(\tilde{\theta}|X)$, so it appears that the HMC sampler converged.

Similarly, the traceplots, $\hat{R}$ values, and effective sample size for $g(\tilde{\theta}|X_{test})$ indicate that the HMC sampler converged. The $\hat{R}$ values are between 0.999 and 1.002 and the effective sample sizes are greater than 1928. Since the X_{test} values are over an evenly spaced grid over the entire range of X, we will use the $\hat{R}$ and effective sample size values for X_{test} to assess the convergence of $g(\theta|X)$.

Since usually the true value of K is not known, we would like to see what happens when Stan is given a larger value of K. In particular, we repeat the above situation, but instead, tell the Stan model that $K = 3$. Figure 4.4 contains the traceplots for θ, which still have a label switching problem but don't appear to converge as well. Figure 4.5 contains the traceplots for $g(\tilde{\theta}|X_{test})$ for 6 random values of X_{test} and figure 4.6 contains boxplots of the $\hat{R}$'s and effective sample sizes for $g(\tilde{\theta}|X_{test})$. Both figures 4.5 and 4.6 indicate that the HMC sampler converged.

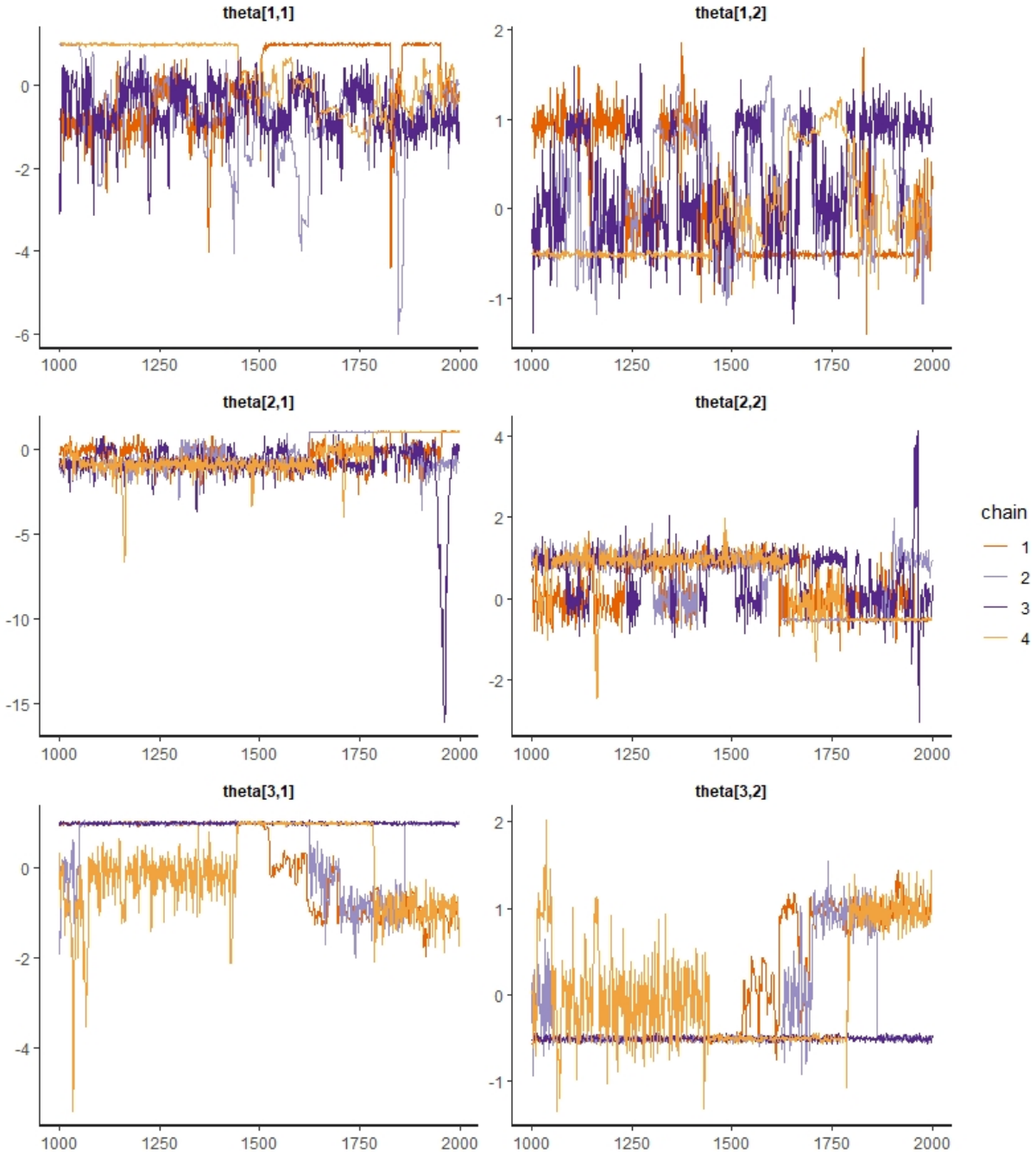

Figure 4.4: The traceplots of θ when $K = 3$ for 4 chains each with 1000 samples after 1000 burnin samples.

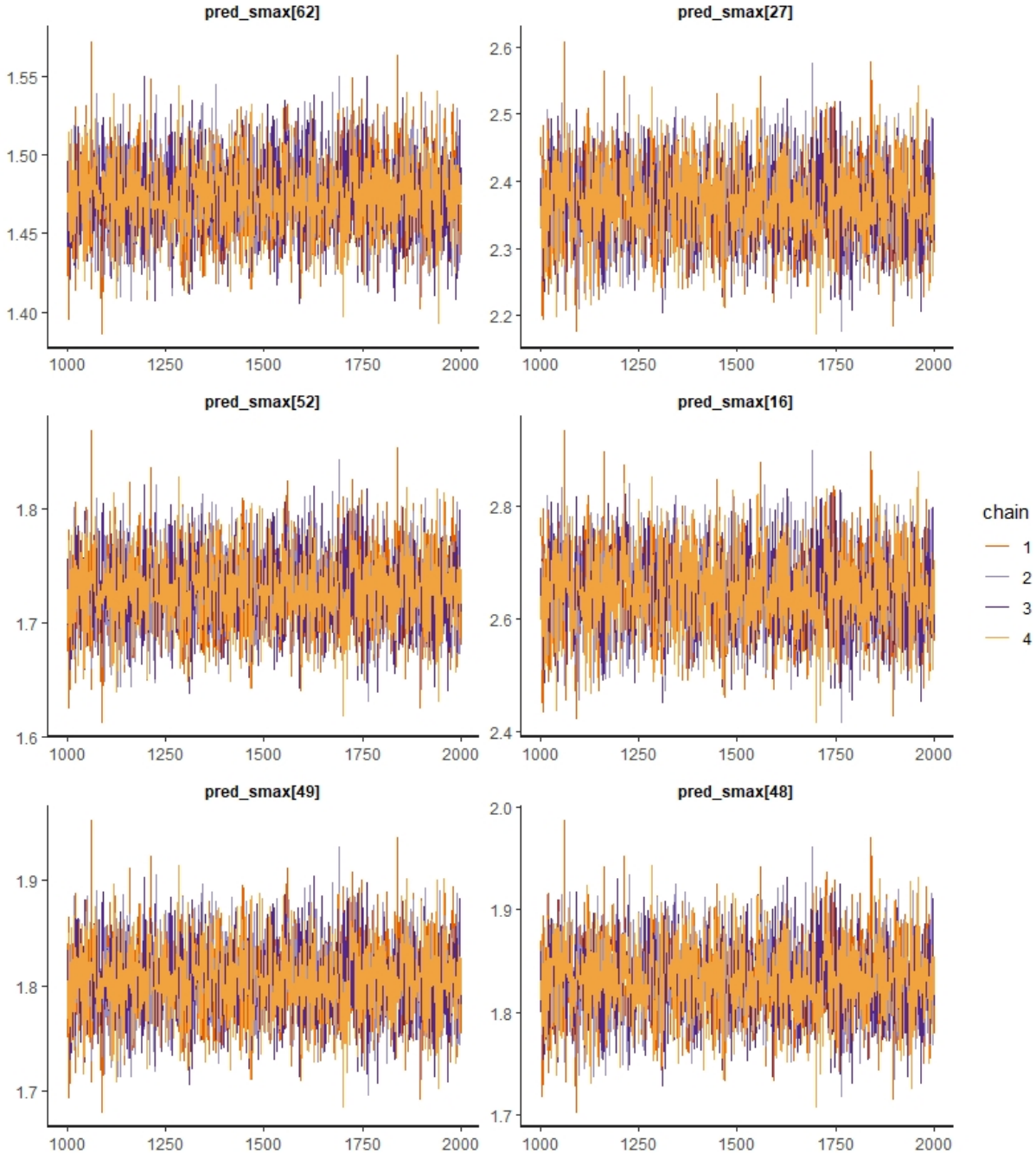

Figure 4.5: The traceplots of $g(\tilde{\theta}|X_{test})$ when $K = 3$ for 4 chains each with 1000 samples after 1000 burnin samples.

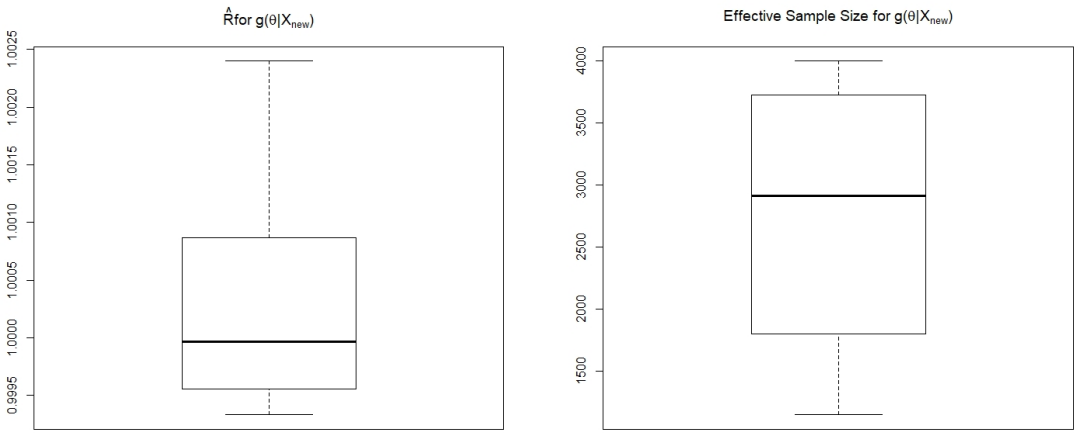

Figure 4.6: Boxplots of the values of $\hat{R}$ (left) and effective sample size (right) of $g(\tilde{\theta}|X_{test})$ when $K = 3$ for 4 chains each with 1000 samples after 1000 burnin samples.

4.4 Simulations

4.4.1 Synthetic Regression Problem

The goal of this simulation is to compare the performance of the SMA regression with other convex regressions. In particular, we will compare it with CAP and Fast CAP from Hannah and Dunson [2013]. Due to the inability to find a public implementation for CAP, we will mimic the first synthetic regression problem in section 6.1.1 of Hannah and Dunson [2013]. In particular, the simulation compares the MSE and the run time for various regression and convex regression methods. MBCR also performed a simulation on the same synthetic regression problem in section 5.1 of Hannah and Dunson [2011], but it is unclear if the simulation setup is the same. The simulation with MBCR did not provide run times, and it was only performed for samples sizes less than or equal to 1000. We will provide the MSE values for MBCR, but we cannot make direct comparisons to these numbers, only general comparisons.

Assume that $X \in \mathbb{R}^5$ has independent standard normal distributions, $y = (X_1 + 0.5X_2 + X_3)^2 - X_4 + 0.25X_5^2 + \epsilon$, and ϵ is standard normally distributed. For each sample size, all methods were run on 10 training datasets and compared with one test dataset, and only allowed to run for 90

minutes. For the SMA regression, the test dataset was half of the size of the training dataset. For each training dataset, the value of K ranged from 2 to 11 by increments of 1. The HMC sampler had four chains, each with 1000 burnin samples and 1000 after burnin samples.

Mean Square Error

Method	$n = 100$	$n = 200$	$n = 500$	$n = 1000$	$n = 2000$	$n = 5000$	$n = 10000$
SMA	0.6439	0.4362	0.2350	0.1270	0.0725	0.0611	0.0428
CAP	1.5884	0.6837	0.2740	0.1644	0.0927	0.0629	0.0450
Fast CAP	1.8661	0.7471	0.3197	0.1526	0.1356	0.0724	0.0566
LSE	15.8340	9.5970	18.0701	9862.4602	–	–	–
Tree	12.2794	9.8356	6.7607	5.3478	4.1230	2.9173	2.3152
GP	8.5056	13.5495	6.8472	3.7610	2.2928	1.2058	–
MARS	8.3517	8.0031	6.8813	6.2618	5.9809	5.8558	5.8234
MBCR	1.0373	0.3679	0.2784	0.2180			

Mean Runtime (sec)

Method	$n = 100$	$n = 200$	$n = 500$	$n = 1000$	$n = 2000$	$n = 5000$	$n = 10000$
SMA	17.46	42.68	187.06	608.39	1140.95	2912.76	3336.23
CAP	0.15	0.24	0.78	1.34	2.18	4.33	9.31
Fast CAP	0.04	0.07	0.15	0.30	0.57	1.14	2.06
LSE	1.56	10.17	226.20	2602.20	–	–	–
Tree	0.06	0.02	0.04	0.09	0.19	0.49	1.15
GP	0.22	0.35	1.35	5.07	22.03	248.72	–
MARS	0.22	0.34	0.76	1.81	3.95	16.65	59.19

Table 4.1: Table from Hannah and Dunson [2013] with the addition of the SMA regression information and the MBCR information from Hannah and Dunson [2011]. The average mean squared error on the test dataset and the average runtime is reported for the SMA regression, convex adaptive partitioning (CAP), Fast CAP, least squares estimator (LSE), Gaussian processes (GP), multivariate adaptive regression splines (MARS), tree regression, and Multivariate Bayesian Convex Regression (MBCR).

Table 4.1 contains the average MSEs and the average run time for each regression method. The SMA MSE value is the average MSE of the model chosen using predictive cross validation, only looking at models that ran for less than 90 minutes. The SMA regression appears to do equally well or slightly better than CAP and Fast CAP, comparable to MBCR, and much better than the other

regression methods. The average run time for the SMA regression method is the average of the run times for the model chosen using minimum predictive cross validation. Throughout the entire simulation, there were only 4 instances where a model ran for more than 90 minutes, as indicated by "–", none of which were SMA. MBCR did not report a run time in their simulation, but SMA was able to run on a larger dataset than MBCR. MBCR only ran on samples up to $n = 1000$, as indicated by the blank spaces. The SMA regression sampler was run using `R` while the other models were all run in Matlab.

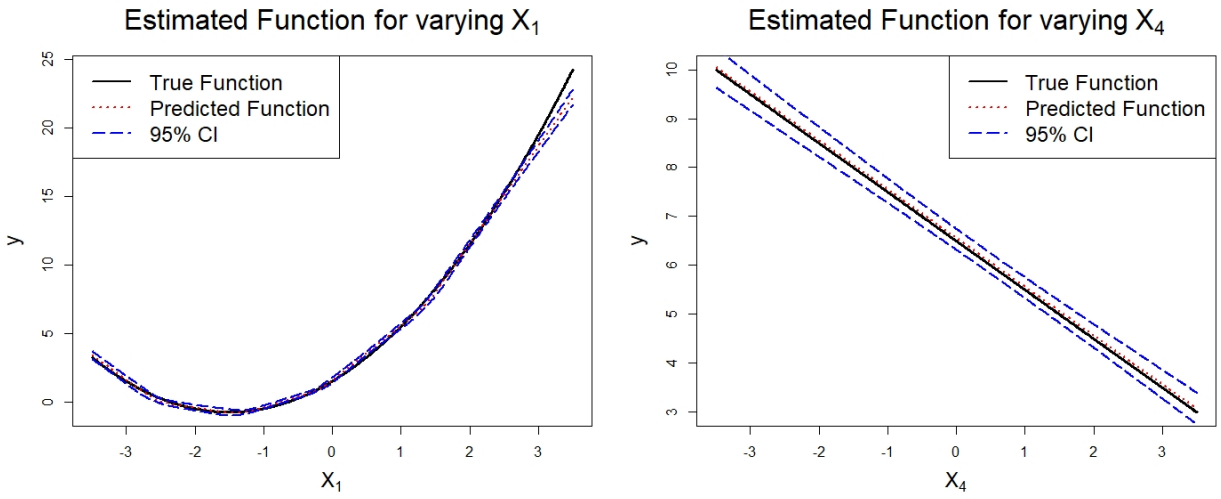

Figure 4.7: True and predicted functions with 95% pointwise credible intervals varying X_1 (left) and X_4 (right) while the other values of X remain constant at 1.

Despite the long run time of the SMA regression, it easily provides posterior predictions for the true function and posterior credible intervals, which most of the other algorithms cannot do. Using the last run of the simulation ($N = 10000$, $K = 11$, trial = 10), figure 4.7 contains plots of the true function (black line), predicted function (red line), and 95% pointwise credible intervals (blue lines) for the predicted function. The left plot varies X_1 between -3.5 and 3.5 and holds the other X values constant at 1. The right plot varies X_4 between -3.5 and 3.5 while holding the other X values constant at 1. The posterior mean of the SMA regression function estimates the true

function when X_1 is varying fairly well with narrow pointwise 95% credible intervals. There are sections where the true function does not lie in the credible interval, but the posterior mean is still very close to the true function.

Figure 4.8 contains a plot of the distance between the true function and the predicted function, $\hat{y}_{test}$, for each of the X_{test} values (left) and a plot of the distance between the true function and the predicted function for the first 10 X_{test} values and their 95% pointwise credible intervals (right). The distance between the true function and the predicted function is usually close to and centered around 0, but the large differences in distance are when the predicted function underestimates the true function. Points below the gray line are over-estimating the true function while points above the gray line are under-estimating the true function. About 68% of the pointwise credible intervals contain the true function value, with about 14% of the credible intervals under-estimating the true function and about 17% of the credible intervals over-estimating the true function.

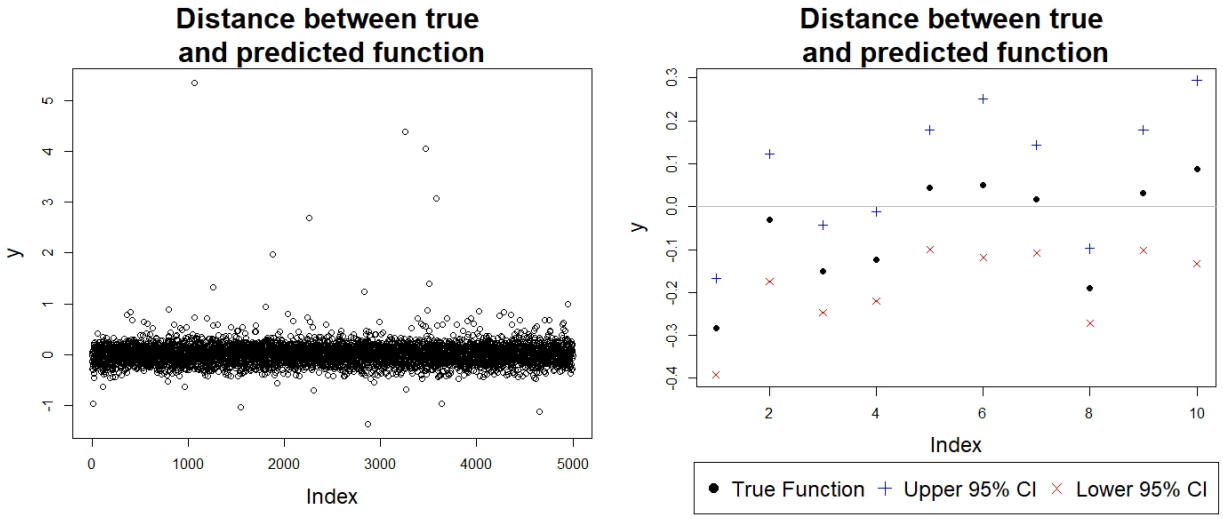

Figure 4.8: Difference between true and predicted functions for all samples in X_{test} (left) and for the first 10 samples in X_{test} with the addition of 95% pointwise credible intervals (right).

In order to see how performance of the SMA regression varies across trials, we re-ran the simulation with 100 trials for $n = 100, 200,$ and 500. Figure 4.9 contains boxplots of the MSE (left)

and predictive cross validation (right) over the 100 trials. Both plots indicate that as n increases, the predicted function gets closer to the true function. The boxplots also indicate that the predictions are fairly stable across replicates.

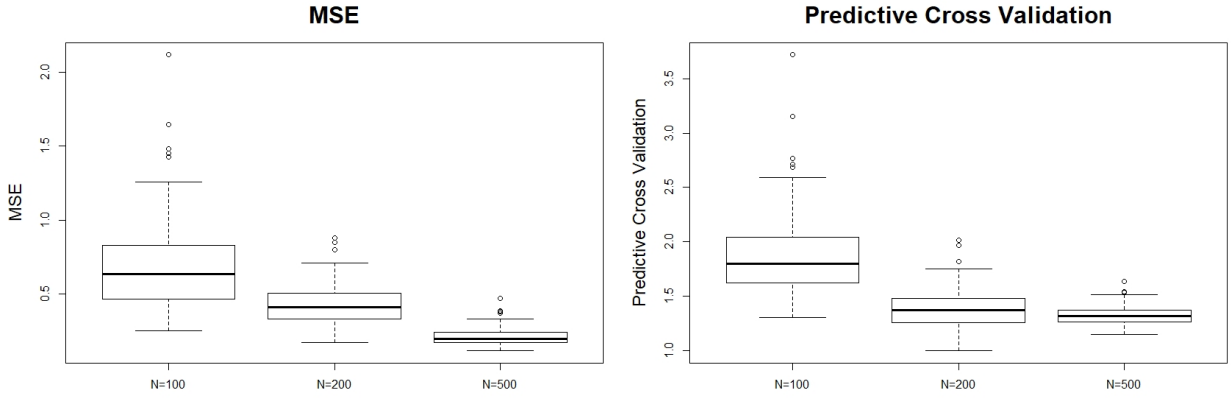

Figure 4.9: Boxplots of the MSE (left) and predictive cross validation (right) over 100 trials.

4.4.2 How to choose β and K

The goal of this simulation is to determine how to choose the values of β and K. We will look at both the predictive cross validation and the marginal log likelihood using the posterior samples of $g(\theta|X)$. Let the number of observations $N = 800$, draw $X \in \mathbb{R}$ from a standard normal distribution, and append a 1 to the front of X. Let the true θ be the same as in equation 4.3. Draw the error ϵ from a normal distribution, $\epsilon \sim N(0, 0.5^2)$, and set $y = \max_{1 \leq i \leq K}\{\theta_i X\} + \epsilon$. Let the hyperparameter β take the values $\{1, 3, 5, 7, 10\}$ and let K range from 1 to 3.

Sample a test dataset on a grid of X values where X ranges from -4 to 4 by increments of 0.05 to be used throughout the entire simulation. For a single trial, re-sample a new training dataset of size 800. For each value of K and β, run the HMC sampler on the training dataset with 4 chains, each with 1000 burnin and 1000 after burnin samples. Use the 4000 posterior samples of $g(\tilde{\theta}|X_{train})$ to estimate the log marginal likelihood using the corrected arithmetic mean estimator

with importance sampling [Pajor et al., 2017]. Then, using the posterior samples of θ and the test dataset, calculate the predictive cross validation, $\|y_{test} - \hat{y}_{test}\|_2^2$. Do this for 30 trials.

Figure 4.10 shows, for each combination of β and K, plots of the true $\max_{1 \leq i \leq 2}\{\theta_i X\}$ function (solid black line) and the 30 posterior means of the predicted $g(\theta|X_{test})$ function (dashed blue lines) from the 30 trials. Clearly, β has no impact when $K = 1$ and $K = 1$ does a bad job of estimating the true function. There does not appear to be much difference between the $K = 2$ and $K = 3$ cases. As β gets larger, the SMA regression does a better job of estimating the true function, especially at the point. However, the SMA regression is not able to exactly match the point because it is a smooth function.

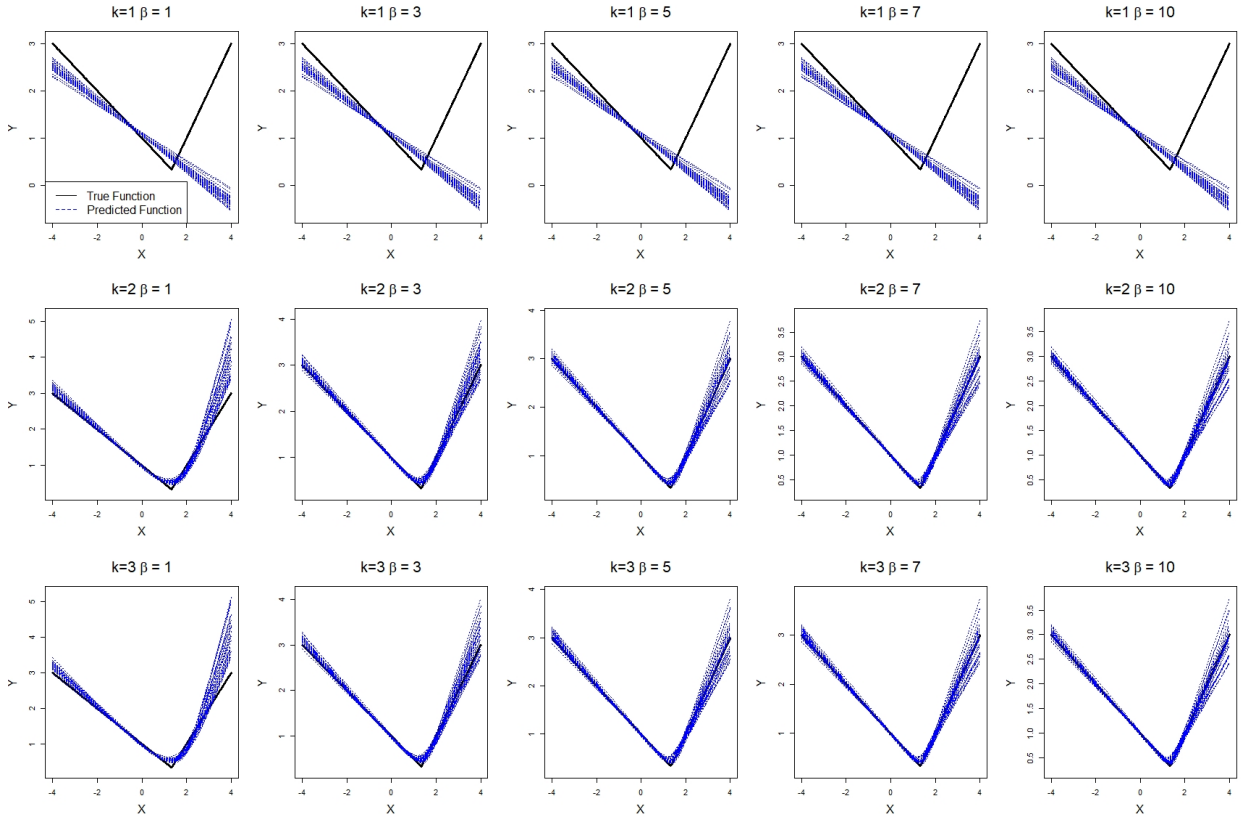

Figure 4.10: Plots of the true function and predicted $g(\theta|X_{new})$ value for each of 30 trials. The true function is the solid black line and the predicted values are the dashed blue lines.

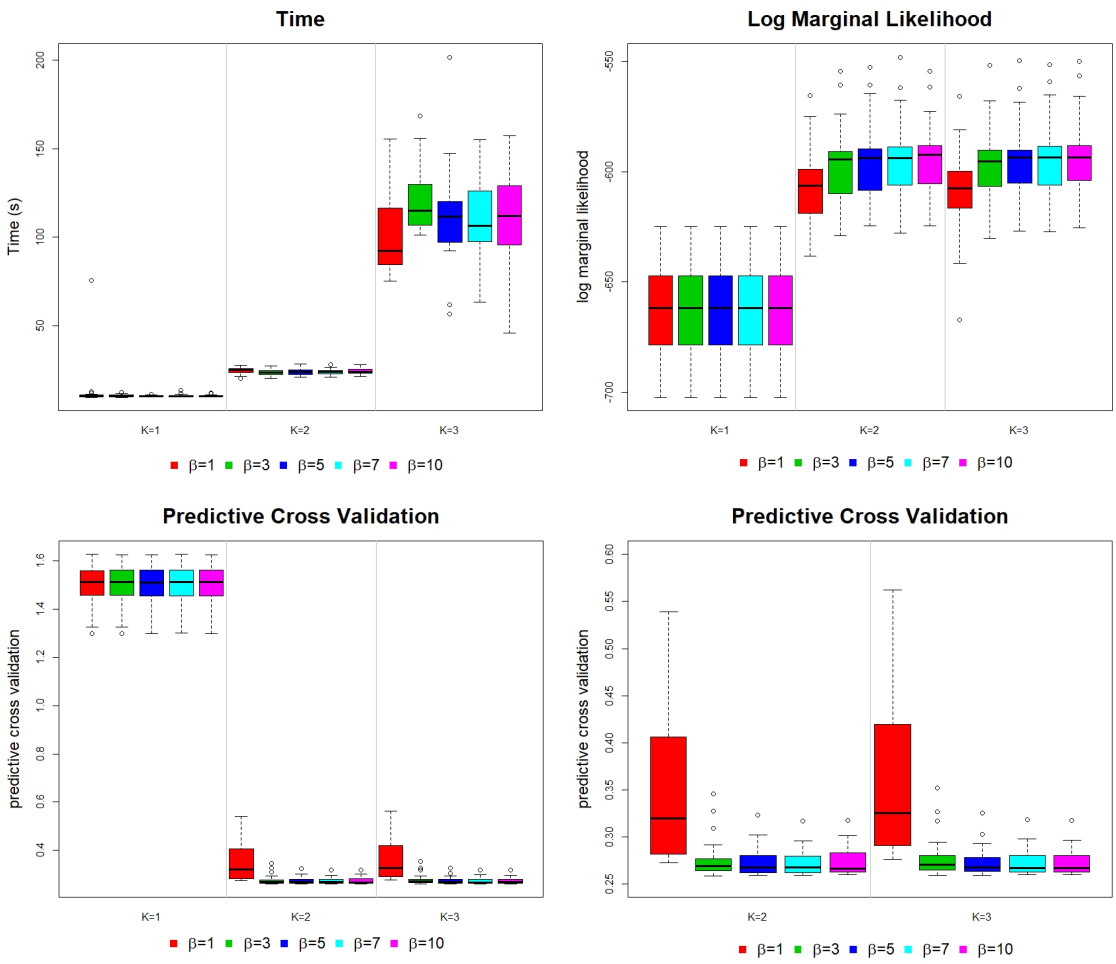

Figure 4.11: Boxplots of the time in seconds for Rstan to run (top left), log marginal likelihood (top right), and predictive cross validation (bottom) for each value of β and each value of K. The left plot of the predictive cross validation contains all values of K and all outliers, while the right plot of the predictive cross validation contains only $K = 2$ and $K = 3$ and does not contain the outliers from the $K = 2$ and $\beta = 1$ case.

Figure 4.11 contains boxplots for the time in seconds for `rstan` to run (top left), the log marginal likelihood (top right), the predictive cross validation (bottom left), and a zoomed in predictive cross validation (bottom right) over the 30 trials for each combination of β and K. As the value of K increases, the time for `rstan` to run increases, but the time does not appear to depend on β. The log marginal likelihood appears to level out at $K = 2$, which is the true value of K. Table 4.2 indicates how many times each value of K had the maximum log marginal likelihood in a trial for each value of β. So while the boxplot indicates that $K = 2$ should be chosen, within a

Maximum Log Marginal Likelihood

	$K=2$	$K=3$
$\beta=1$	20	10
$\beta=3$	13	17
$\beta=5$	12	18
$\beta=7$	11	19
$\beta=10$	12	18

Table 4.2: Table of how many times each value of K maximized the log marginal likelihood for each value of β.

Predictive Cross Validation

	$K=2$	$K=3$
$\beta=1$	30	0
$\beta=3$	22	8
$\beta=5$	20	10
$\beta=7$	16	14
$\beta=10$	12	18

Table 4.3: Table of how many times each value of K minimized the predictive cross validation for each value of β.

single trial, both $K=2$ and $K=3$ will be chosen with $K=3$ chosen more often. The predictive cross validation also levels out at $K=2$ and when looking at the zoomed in predictive cross validation plot, it appears that the predictive cross validation for $K=2$ and $K=3$ are similar. Table 4.3 indicates how many times each value of K had the minimum predictive cross validation in a trial for each value of β. Within a trial, both $K=2$ and $K=3$ will be chosen with $K=2$ chosen more often. As β increases, the amount of times $K=3$ is chosen increases.

Figure 4.12 contains boxplots for the median effective sample size (left) and median $\hat{R}$ (right) for each value of K and β. The median is over the 161 effective sample size values and $\hat{R}$ values calculated from $g(\tilde{\theta}|X_{test})$ and the boxplots are over the 30 median values for each trial. As K increases, the range of values the median effective sample size takes increases. It appears that for $K=2$, as β increases, the median effective sample size increases, but for $K=3$, as β increases, the median effective sample size appears to decrease. The β does not appear to have an effect on the median $\hat{R}$ value. When $K=3$, the median $\hat{R}$ has more outliers, but otherwise the value of K doesn't appear to have an effect on the median $\hat{R}$ value. Both the median effective sample size and the median $\hat{R}$ values indicated convergence of the sampler.

In order to choose K, we will look at 3 different methods. We will look at the value of K chosen using predictive cross validation and the value of K that maximizes the log marginal likelihood. We will also look at a model average that does not choose a specific value of K and instead averages the predicted y values over K, weighted by the marginal likelihoods. In particular, let m_k

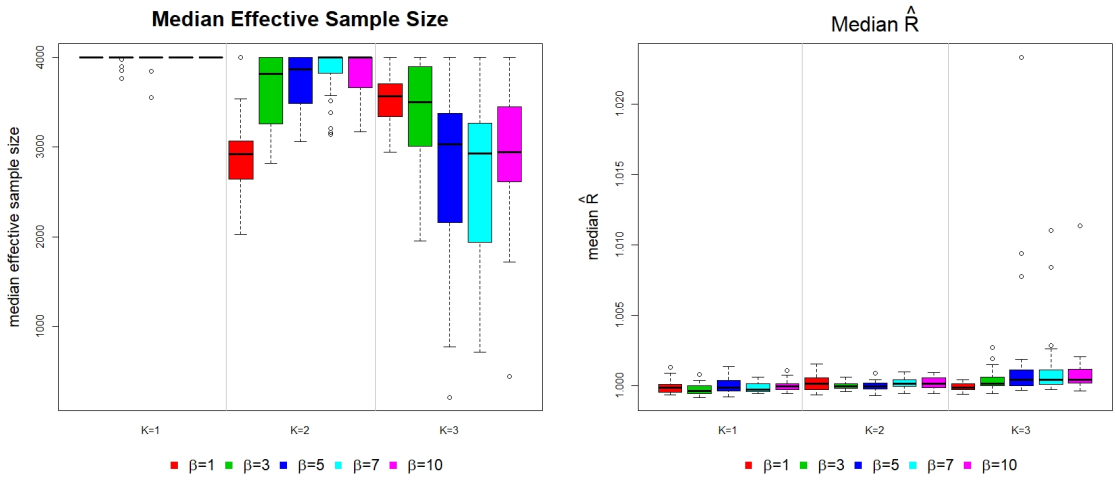

Figure 4.12: Boxplots of the median effective sample size and $\hat{R}$ values for each value of β and K where the median is over the range of values for $g(\tilde{\theta}|X_{test})$ and the boxplots are over the trials

be the log marginal likelihood for the model using k components and $\hat{y}_{test}^k$ be the $\hat{y}_{test}$ value for the model using k components. Then

$$\hat{y}_{avg} = \sum_{i=1}^{K} \frac{e^{m_i}}{\sum_{j=1}^{K} e^{m_j}} \hat{y}_{test}^i$$

is the predicted y value for the model average. Let $\hat{y}_{mod}$ be one of three values, where the subscript mod indicates how K was chosen. It can be the $\hat{y}_{test}$ value for the model chosen using predictive cross validation, the $\hat{y}_{test}$ value for the model with the maximum log marginal likelihood, or $\hat{y}_{avg}$. The three methods will be compared using the predictive cross validation, $\|y_{test} - \hat{y}_{mod}\|_2^2$, and the MSE, $\|\max_{1\leq i\leq K}\{\theta_i X\} - \hat{y}_{mod}\|_2^2$ which is the MSE between the true function and the estimated function evaluated at X_{test}. Usually, the MSE cannot be known because it requires knowledge of the true function. However, we are interested in how well the SMA regression estimates the true function, so we will look at the MSE to help determine how to best choose K.

Figure 4.13 contains boxplots of the MSE and the predictive cross validation for each method of estimating K and for each value of β. The three methods appear to do equally well and all values of β greater than 1 appear to do equally well. Since the value of β does not seem to

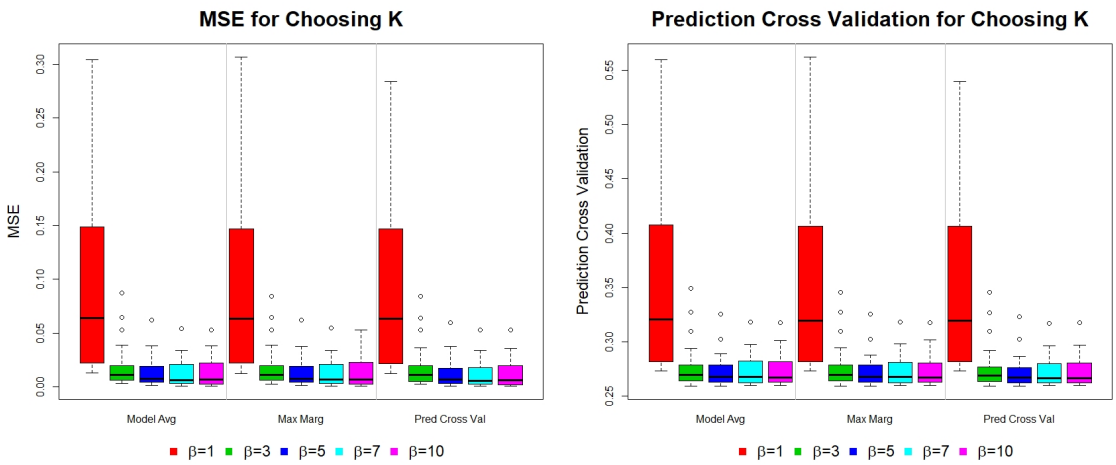

Figure 4.13: Boxplots over β of the MSE (left) and predictive cross validation (right) for the model averaging and the models chosen from the maximum log marginal likelihood and from the predictive cross validation.

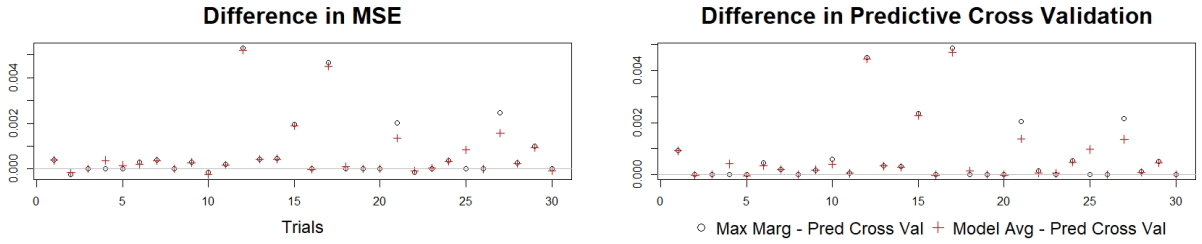

Figure 4.14: Plot of the difference in MSE (left) and predictive cross validation (right) over each trial for $\beta = 10$ and taking the difference with the model selected by minimizing the predictive cross validation.

affect the convergence or accuracy of the sampler and we are using the softmax function as an approximation to the pointwise maximum of affine functions, we recommend using $\beta = 10$.

Figure 4.14 contains plots of the difference in MSE (left) and predictive cross validation (right) over the 30 trials for the $\beta = 10$ case. The differences are the maximum log marginal likelihood method minus the predictive cross validation method (black circles) and the model averaging method minus the predictive cross validation method (red plus sign). A point greater than the gray line indicates that the model chosen by the predictive cross validation method had a lower MSE

or predictive cross validation value. The model chosen by the predictive cross validation usually (but not always) had the lowest MSE and always had the lowest predictive cross validation value. There were instances where the model chosen by maximizing the log marginal likelihood or model averaging shared the lowest predictive cross validation value.

The differences between the three methods for both the MSE and predictive cross validation are very small, so all three methods could be used in this situation. However, figure 4.15 contains boxplots for the log marginal likelihood of each value of K at 3 different sample sizes for the synthetic regression simulation in subsection 4.4.1. The boxplots for the log marginal likelihood at all of the sample sizes have the same downward trend, so it appears that the marginal likelihood does not estimate the number of components K well, perhaps due to the large number of parameters in that problem.

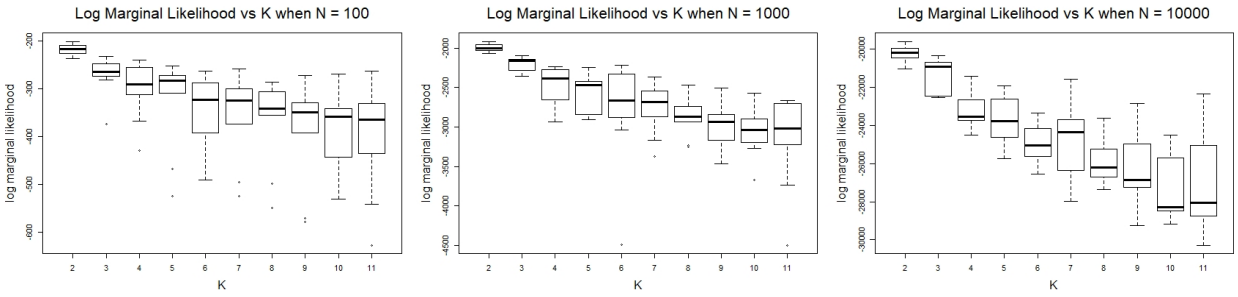

Figure 4.15: Boxplots of the log marginal likelihood for each value of K for 3 sample sizes.

4.5 Discussion

In this section, we have presented a HMC algorithm for estimating a convex function using the softmax function, where the softmax function is used as a smooth approximation to the maximum of affine functions. It appears that the SMA regression does a good job of estimating various convex functions. Three methods were presented to help choose the number of components K. It appears that the predictive cross validation method consistently performs well, while the maximum log marginal likelihood method, and therefore the model averaging method, can perform well, but

does not consistently perform well. Therefore, we suggest using the predictive cross validation method to choose the number of components K.

5. FACTOR MODELS FOR COUNT DATA: AN EXPLORATION OF THE COVARIANCE STRUCTURE

5.1 Introduction

Latent factor models [Bartholomew, 1987] are highly popular for structured dimension reduction and dependence modeling in a variety of fields such as genomics, finance, and psychometrics to name a few. In their most general description, factor models postulate that multivariate observations are conditionally independent given one or more unobserved factors, and dependence among the variables are induced upon marginalization over the distribution of the latent factors. The number of factors is typically much smaller than the ambient dimension of the observations, leading to a parsimonious model for the dependence structure.

For continuous data, a linear Gaussian factor model is most commonly used, which linearly relates the observations to the latent factors, up to additive Gaussian noise, via a tall-and-skinny factor loadings matrix. A Gaussian latent factor distribution is commonly assumed, which, upon marginalization, implies a decomposition of the observational covariance as the sum of a low-rank matrix and a diagonal matrix. Such a decomposition of the covariance succinctly conveys the parsimony implied by latent factor models; the effective number of parameters is reduced from quadratic to linear in the number of variables. When the number of variables is comparable or even larger than the sample size, further dimension reduction is often achieved by assuming the factor loadings to be sparse [West, 2003]. Various extensions of the Gaussian factor model for continuous data include heavy-tailed distributions for the latent factors and/or the additive noise [Ando, 2009], mixtures of factor analyzers [McLachlan and Peel, 2000], and non-linear factor models [Lawrence, 2004].

Factor models can accommodate mixed data types through an underlying Gaussian latent factor structure [Muthen, 1983], where each observation is linked to a continuous latent variable, with a Gaussian factor model for the collection of continuous latent variables. For example, binary data

$y_{ij} \in \{0, 1\}$ can be assumed to arise through thresholding of a latent continuous variable z_{ij}, with a joint Gaussian model for the z_{ij}s leading to the multivariate probit model [Ashford and Sowden, 1970, Chib and Greenberg, 1998, Ochi and Prentice, 1984]. Extensions to nominal data with more than 2 categories is possible through introducing a vector of latent variables [Aitchison and Bennett, 1970, Zhang et al., 2008] per observation. A related strategy is to assume an appropriate exponential family model for each of the individual outcomes, with a joint Gaussian factor model on the canonical parameters [Sammel et al., 1997, Moustaki and Knott, 2000, Dunson, 2000, 2003]. For binary or nominal data, a different class of factor models constitute of a low-rank probabilistic decomposition of the joint probability mass function of the observed variables [Dunson and Xing, 2009, Bhattacharya and Dunson, 2012, Zhou et al., 2015]

In this section, we focus on factor models for multivariate count data. Such data routinely appear as read counts in next-generation sequencing for micro-RNA. This is a method of measuring gene expression by using the number of reads from a sample [Lee et al., 2013]. Accordingly, there is increased interest in modeling high-dimensional count data using latent variable models [Wedel et al., 2003, Lee et al., 2013]. Multivariate count data also appear in numerous other applications. For example, the number of accidents sustained by shunters over set time periods [Aitchison and Ho, 1989] and the number of views a TV channel has over various households [Wedel et al., 2003]. A common method for modeling multivariate count data is through the multivariate Poisson-log normal model [Aitchison and Ho, 1989]. A d-dimensional multivariate Poisson-log normal distribution is hierarchically specified by assuming the d variables follow independent Poisson distributions, and then assigning a multivariate log-normal distribution to the vector of Poisson means. A factor structure can be readily incorporated by linearly relating the log means to lower-dimensional Gaussian latent factors [Zhou et al., 2012a, Wedel et al., 2003, Lee et al., 2013]. The negative binomial (NB) distribution is another popular choice for count data modeling which, unlike the Poisson distribution, allows for over-dispersion [Lawless, 1987]. Recent applications of the NB distribution to high-dimensional and/or latent variable models include Klami et al. [2015], Zhou and Carin [2015], Zhou et al. [2012b].

Although frequently used in practice, there is little understanding of the dependence structure induced by such Poisson and NB factor models. For example, the natural question of whether the covariance structure continues to admit a near low-rank decomposition as in the continuous case seems unexplored. With such motivation, we first derive expressions for the covariance matrices of Poisson and NB factor models. Although the covariance in either case can be represented as the sum of a diagonal matrix and a non-negative definite matrix, we prove that the non-diagonal part has full rank. This seemingly suggests a significant departure from the continuous case where the effect of dimensional reduction directly shows up in the decomposition of the covariance matrix.

The observations above are somewhat expected. The rank is a measure of *linear* dependence and hence interacts nicely with linear dimension reduction in Gaussian factor models. However, as one goes beyond Gaussianity to general exponential family models and beyond, non-linearities in the link functions distort the linear geometry, and hence the effect of dimension reduction do not reflect in the rank. Our main contribution is to consider a class of continuous relaxations of the rank, known collectively as *effective rank*, and exhibit theoretically and numerically that the effect of dimension reduction is reflected in the effective rank.

The rest of the section is organized as follows. In subsection 5.2, we introduce some notation. We also introduce the effective rank and some if its properties. In subsection 5.3, we introduce both the Poisson and the Negative Binomial factor models. Subsection 5.4 introduces the main theorem that the covariance matrices of the Poisson and NB factor models are the sum of an effectively low rank matrix and a diagonal matrix, while subsection 5.5 is a proof of the theorem. Subsection 5.6 contains a simulated example that the bounds produced by our theorem are conservative bounds. We conclude with a discussion in subsection 5.7.

5.2 Preliminaries

We introduce some basic notation used throughout Section 5. For a square matrix A, $\mathrm{tr}\,(A)$ denotes its trace. We use I_d to denote the $d \times d$ identity matrix, while $\mathrm{J}_d = 11^{\mathrm{T}}$ denotes a $d \times d$ matrix of ones. For an $m \times r$ matrix A (with $m > r$), $s_i(A) := s_i = \sqrt{\nu_i}$ for $i = 1, \ldots, r$ denote the singular values of A, where $\nu_1 \geq \nu_2 \geq \ldots \geq \nu_r \geq 0$ are the eigenvalues of $A^{\mathrm{T}}A$. The largest

singular value $s_{max}(A) := s_1(A)$ is the operator norm of A, which shall also be denoted $\|A\|_2$. Similarly, $s_{min}(A)$ shall denote the smallest singular value.

For matrices A and B of the same dimension, $A \circ B$ denotes the Hadamard or Schur product of A and B, with $(A \circ B)_{ij} = A_{ij}B_{ij}$. We use $\exp_\circ$ to denote the Hadamard exponential function, i.e., $\big(\exp_\circ(A)\big)_{ij} = \exp(A_{ij})$. We shall make frequent use of Schur's theorem [Horn and Johnson, 1985] which states that for non-negative definite (n.n.d.) matrices A and B, $A \circ B$ is also n.n.d., and is in particular positive definite (p.d.) when both A and B are so. An important consequence is that $\exp_\circ(A)$ is p.d. for any n.n.d. matrix A.

Let $\mathcal{N}_d(\mu, \Sigma)$ denote the d-dimensional normal distribution with mean $\mu \in \mathbb{R}^d$ and covariance matrix Σ. We use the shape-rate formulation of the Gamma distribution, so that Gamma(α, β) has mean α/β. The negative binomial distribution NB(r, p), with parameters $r > 0$ and $p \in (0, 1)$, has probability mass function

$$\frac{\Gamma(k + r)}{\Gamma(k + 1)\Gamma(r)}\, p^k (1 - p)^r\, \mathbb{1}_{(0,1,\ldots)}(k). \tag{5.1}$$

The NB distribution can be expressed as a Poisson-Gamma mixture. If $y \mid \lambda \sim$ Poisson(λ) and $\lambda \sim$ Gamma$(r, (1 - p)/p)$, then $y \sim$ NB(r, p). Clearly, $\mathbb{E}(y) = rp/(1 - p)$ and var$(y) = rp/(1 - p)^2$.

5.2.1 Effective rank

The rank $r(\cdot)$ of a matrix is a measure of *linear* dependence between its columns (or equivalently, rows), which is equivalently the vector space dimension of its column (or row) space. When there is an approximate linear relationship, i.e., the column space is close to being lower dimensional, the rank fails to capture this. As a somewhat extreme example, consider the compound symmetry covariance matrix, $\Sigma_\rho = (1 - \rho)\mathrm{I}_d + \rho\, \mathrm{J}_d$ for $\rho \in [0, 1]$. For any $\rho \in [0, 1)$, Σ_ρ is positive definite, and hence full rank, while the rank drops to one when $\rho = 1$. As ρ gets closer to one, Σ_ρ becomes increasingly close to being rank deficient. However this is not captured in the rank, which jumps from d to 1 at $\rho = 1$. A more robust measure of the intrinsic dimensionality should instead smoothly change with small perturbations in the matrix. We discuss two such measures below.

The first measure is only defined for non-negative definite matrices. Since we are concerned with covariance matrices, this is not restrictive for our purpose. For a $d \times d$ non-negative definite matrix A, its *effective rank* [Vershynin, 2010] is:

$$r_e(A) := \frac{\mathrm{tr}\,(A)}{\|A\|_2} = \frac{\sum_{j=1}^{d} s_j(A)}{s_1(A)} = 1 + \frac{\sum_{j=2}^{d} s_j(A)}{s_1(A)}, \tag{5.2}$$

where the $s_j(A)$s are the singular values, or equivalently eigenvalues (since A is p.d.), of A, in decreasing order. It is immediately apparent from the definition that $1 \leq r_e(A) \leq d$ and $r_e(cA) = r_e(A)$ for any $c > 0$. Also, if $r(A) = 1$, then $r_e(A) = 1$, and $r_e(\mathrm{I}_d) = d$, so that both bounds are achieved. It can be additionally proved from the Cauchy–Schwartz inequality that $r_e(A) \leq r(A)$ for any A, so that the effective rank always provides a lower bound to the actual rank. Further, this bound is sharp as for any $r \in \mathbb{Z}$, $1 \leq r \leq d$, there exists a matrix A with $r(A) = r_e(A) = r$. Finally, and importantly for our purpose, $r_e(A)$ is a smooth function of A, which can be quantified as follows: given n.n.d. matrices A and B, and for any $\epsilon > 0$, there exists $\delta > 0$ such that if $\|A - B\|_2 \leq \delta$, then $|r_e(A) - r_e(B)| \leq \epsilon$. Finally, we comment that the expression for $r_e(A)$ involves the reciprocal of the condition number $\kappa(A) := s_{max}(A)/s_{min}(A)$. The condition number is commonly used as a measure of numerical stability of matrices in numerical linear algebra [Golub and Van Loan, 2012].

Our second notion of intrinsic dimensionality, also called effective rank in Roy and Vetterli [2007], applies more generally to non-square matrices. We however restrict ourselves to square matrices in the discussion below. Let $p_j = s_j(A)/\sum_{i=1}^{d} s_i(A)$ for $j = 1, \ldots, d$, so that $\mathrm{p} = (p_1, \ldots, p_d)$ is the vector of singular values normalized to the probability simplex. Following Roy and Vetterli [2007], define

$$\widetilde{r}_e(A) = \exp\{\mathcal{H}(\mathrm{p})\}, \tag{5.3}$$

where $\mathcal{H}(\mathrm{p}) = -\sum_{j=1}^{d} p_j \log p_j$ is the Shannon entropy of the probability vector p. The Shannon entropy is a measure of disorder or randomness of a probability distribution, with higher values

indicating more disorder. In particular, $0 \leq \mathcal{H}(\mathrm{p}) \leq \log d$, so that we again have $1 \leq \widetilde{r}_e(A) \leq d$. Moreover, $\widetilde{r}_e(A) = 1$, or equivalently, $\mathcal{H}(\mathrm{p}) = 0$, if and only if p assigns all its mass to one point, which in the current context means A has only one positive singular value and hence has rank one. On the other hand, $\widetilde{r}_e(A) = d$ implies $\mathcal{H}(\mathrm{p}) = \log d$, which happens if and only if $p_i = 1/d$ for all i, or equivalently $A = \mathrm{I}_d$. Next, since $\widetilde{r}_e(A)$ only depends on the normalized singular values, it immediately follows that $\widetilde{r}_e(cA) = \widetilde{r}_e(A)$ for any non-zero constant c. Roy and Vetterli [2007] additionally prove that $\widetilde{r}_e(A) \leq r(A)$. $\widetilde{r}_e(A)$ inherits various other properties of the usual rank $r(A)$; see Roy and Vetterli [2007] for more details.

Both measures of effective dimensionality, $r_e(\cdot)$ and $\widetilde{r}_e(\cdot)$, are smooth functions of their argument and inherit important properties of the usual rank. Moreover, both measures are bounded above by the usual rank. A plot of $r_e(\cdot)$ and $\widetilde{r}_e(\cdot)$ versus ρ for the matrix Σ_ρ mentioned at the beginning of this subsection is provided in Figure 5.1 for various values of d.

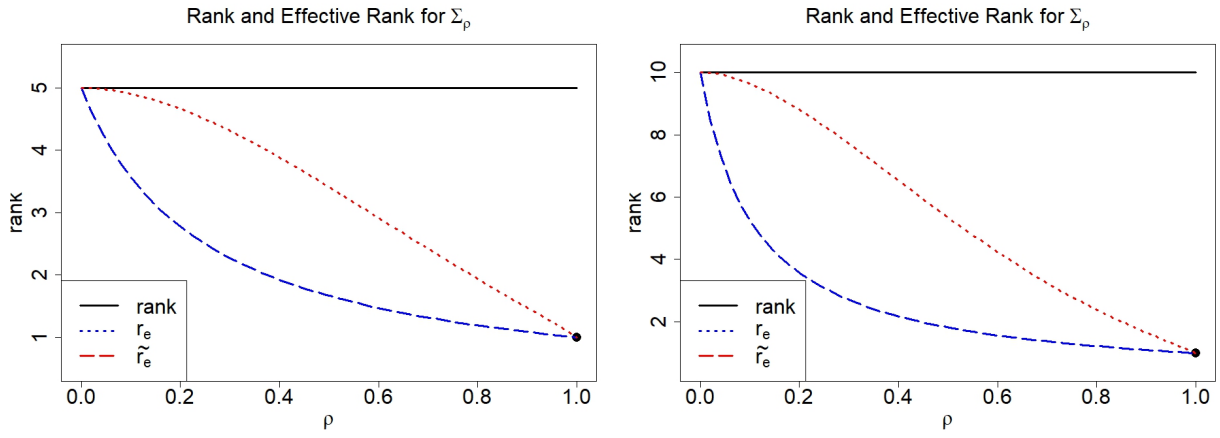

Figure 5.1: Plot of $r(\Sigma_\rho)$, $r_e(\Sigma_\rho)$ and $\tilde{r}_e(\Sigma_\rho)$ for $\rho \in [0, 1]$ and $d = \{5, 10\}$

From the plot, it can be seen that $\widetilde{r}_e(\Sigma_\rho)$ is always above $r_e(\Sigma_\rho)$. This is not a coincidence, and can be proved generally. This is summarized in Proposition 5.2.1 below.

Proposition 5.2.1. *For any $d \times d$ non-negative definite matrix A, $1 \leq r_e(A) \leq \widetilde{r}_e(A) \leq r(A) \leq d$.*

Proof. The inequalities other than $r_e(A) \leq \widetilde{r}_e(A)$ are already known. We have,

$$\widetilde{r}_e(A) = \exp\left\{ -\sum_{j=1}^{d} p_j \log p_j \right\} \geq \exp(-\log p_1) = \frac{1}{p_1} = r_e(A),$$

where we used that $-\log p_j \geq -\log p_1$ for all j and $\sum_{j=1}^{d} p_j = 1$. □

In Roy and Vetterli [2007], it was proved that for non-negative definite matrices A and B of the same size, $\widetilde{r}_e(A+B) \leq \widetilde{r}_e(A) + \widetilde{r}_e(B)$. We show below that the same conclusion holds for $r_e(\cdot)$, extending the parallel with the usual rank.

Proposition 5.2.2. *For non-negative definite matrices A and B of the same size,*

$$r_e(A+B) \leq r_e(A) + r_e(B).$$

Proof. Since both A and B are n.n.d., for any x with $\|x\| = 1$, $x^{\mathrm{T}}(A+B)x \geq x^{\mathrm{T}}Ax$. Taking the supremum over all x with $\|x\| = 1$, we obtain $\|A+B\|_2 \geq \|A\|_2$. By symmetry, $\|A+B\|_2 \geq \|B\|_2$. We then have,

$$r_e(A+B) = \frac{\mathrm{tr}\,(A+B)}{\|A+B\|_2} \leq \frac{\mathrm{tr}\,(A)}{\|A\|_2} + \frac{\mathrm{tr}\,(B)}{\|B\|_2} = r_e(A) + r_e(B).$$

□

We utilize Proposition 5.2.2 in subsection 5.4 later on.

5.3 Count data factor models

5.3.1 Poisson factor model

The Poisson factor model is as follows:

$$
\begin{aligned}
y_{ij} \mid q_{ij} &\sim Poisson(q_{ij}),\ i = 1, \ldots, n,\ \ j = 1, \ldots, p \\
\log(q_i) &= \Lambda \eta_i \\
\eta_i &\sim \mathcal{N}_k(0, I_k)
\end{aligned}
$$

where $\log(q_i) = [\log(q_{i1}), \ldots, \log(q_{ip})]^T$, Λ is the $p \times k$ fixed but unknown factor loading matrix, and η_i are the latent factors. By marginalizing out η_i, $\log(q_i) \sim \mathcal{N}_k(0, \Lambda\Lambda^T)$. Then, q_i is multivariate log-normally distributed. Thus:

$$
\begin{aligned}
E[y_{ij}] &= e^{\lambda_j^{\mathrm{T}} \lambda_j / 2} \\
\mathrm{Var}(y_{ij}) &= e^{\lambda_j^{\mathrm{T}} \lambda_j / 2} + e^{\lambda_j^{\mathrm{T}} \lambda_j} \left(e^{\lambda_j^{\mathrm{T}} \lambda_j} - 1 \right) \\
\mathrm{Cov}(y_{ij}, y_{ik}) &= e^{(\lambda_j^{\mathrm{T}} \lambda_j + \lambda_k^{\mathrm{T}} \lambda_k)/2} \left(e^{\lambda_j^{\mathrm{T}} \lambda_k} - 1 \right)
\end{aligned}
$$

where $\Lambda = [\lambda_1, \ldots, \lambda_p]^{\mathrm{T}}$ and λ_j is a k-dimensional vector. Note that if $\lambda_j^{\mathrm{T}} \lambda_k = 0$ then $\mathrm{Cov}(y_{ij}, y_{ik}) = 0$, if $\lambda_j^{\mathrm{T}} \lambda_k < 0$ then $\mathrm{Cov}(y_{ij}, y_{ik}) < 0$, and if $\lambda_j^{\mathrm{T}} \lambda_k > 0$ then $\mathrm{Cov}(y_{ij}, y_{ik}) > 0$. Since $\lambda_j^{\mathrm{T}} \lambda_k = \mathrm{Cov}\{\log(q_{ij}), \log(q_{ik})\}$, the sign of $\mathrm{Cov}(y_{ij}, y_{ik})$ is determined by the sign of $\mathrm{Cov}\{\log(q_{ij}), \log(q_{ik})\}$.

Let $\mathrm{Cov}(y_i) = \Omega_Y$. Then Ω_Y can be written in terms of matrices as follows:

$$
\begin{aligned}
w_j &= e^{\lambda_j^{\mathrm{T}} \lambda_j / 2} \qquad w = [w_1, \ldots, w_p]^{\mathrm{T}} \qquad W = w w^{\mathrm{T}} \\
H &= \exp_{\circ}(\Lambda\Lambda^{\mathrm{T}}) \\
\Gamma &= W \circ H - W \circ J \\
\Delta &= \text{diag} \left\{ e^{\lambda_1^{\mathrm{T}} \lambda_1 / 2}, \ldots, e^{\lambda_p^{\mathrm{T}} \lambda_p / 2} \right\} \\
\Omega_Y &= \Gamma + \Delta
\end{aligned}
$$

5.3.2 Negative Binomial factor model

The Negative Binomial factor model is as follows:

$$\begin{aligned} y_{ij} \mid r_j, q_{ij} &\sim NB(r_j, q_{ij}),\ i = 1, \ldots, n,\ \ j = 1, \ldots, p \\ \text{logit}(q_i) &= \Lambda \eta_i \\ \eta_i &\sim \mathcal{N}_k(0, I_k) \end{aligned}$$

where $\text{logit}(q_i) = [\text{logit}(q_{i1}), \ldots, \text{logit}(q_{ip})]^T$, Λ is the $p \times k$ fixed but unknown factor loading matrix, and η_i are the latent factors. By marginalizing out η_i, $\text{logit}(q_i) \sim \mathcal{N}_k(0, \Lambda\Lambda^T)$. Then, $e^{\text{logit}(q_i)}$ is log-normally distributed. Thus, there is an explicit form for the mean and covariance of y_{ij}.

$$\begin{aligned} E[y_{ij}] &= r_j e^{\lambda_j^{\mathrm{T}} \lambda_j / 2} \\ \text{Var}(y_{ij}) &= r_j e^{\lambda_j^{\mathrm{T}} \lambda_j / 2} + r_j e^{2\lambda_j^{\mathrm{T}} \lambda_j} + r_j^2 e^{\lambda_j^{\mathrm{T}} \lambda_j} \left(e^{\lambda_j^{\mathrm{T}} \lambda_j} - 1 \right) \\ \text{Cov}(y_{ij}, y_{ik}) &= r_j r_k e^{(\lambda_j^{\mathrm{T}} \lambda_j + \lambda_k^{\mathrm{T}} \lambda_k)/2} \left(e^{\lambda_j^{\mathrm{T}} \lambda_k} - 1 \right) \end{aligned}$$

where $\Lambda = [\lambda_1, \ldots, \lambda_p]^{\mathrm{T}}$ and λ_j is a k-dimensional vector. Note that the sign of $\text{Cov}(y_{ij}, y_{ik})$ is determined by the sign of $\text{Cov}\{\log(q_{ij}), \log(q_{ik})\}$.

Let $\text{Cov}(Y_i) = \Omega_Y$. Then Ω_Y can be written as follows:

$$\begin{aligned} w_j &= r_j e^{\lambda_j^{\mathrm{T}} \lambda_j / 2} \qquad w = [w_1, \ldots, w_p]^{\mathrm{T}} \qquad W = w w^{\mathrm{T}} \\ H &= \exp_{\circ}(\Lambda\Lambda^{\mathrm{T}}) \\ \Gamma &= W \circ H - W \circ J \\ \Delta &= \text{diag} \left\{ r_1 e^{\lambda_1^{\mathrm{T}} \lambda_1 / 2} + r_1 e^{2\lambda_1^{\mathrm{T}} \lambda_1}, \ldots, r_p e^{\lambda_p^{\mathrm{T}} \lambda_p / 2} + r_p e^{2\lambda_p^{\mathrm{T}} \lambda_p} \right\} \\ \Omega_Y &= \Gamma + \Delta \end{aligned}$$

5.3.3 Both models

The covariance structure for both models can be written as $\Omega_Y = W \circ H - W \circ J + \Delta$, where H is the same for both models, Δ is a diagonal matrix for both models, and the W matrix for the NB model is the W matrix from the Poisson model but with an extra r_j term.

Since $\text{rank}(X \circ Y) \leq \text{rank}(X) * \text{rank}(Y)$ [Styan, 1973], $W \circ J$ is a rank 1 matrix. Thus, if $W \circ H$ is a low-rank matrix, Ω_Y is the sum of a low-rank matrix and a diagonal matrix, which mimics the variance of the Gaussian factor model. However, subsection 5.5 shows that $W \circ H$ is positive definite, and therefore full rank. Thus, we need to explore the covariance structure further.

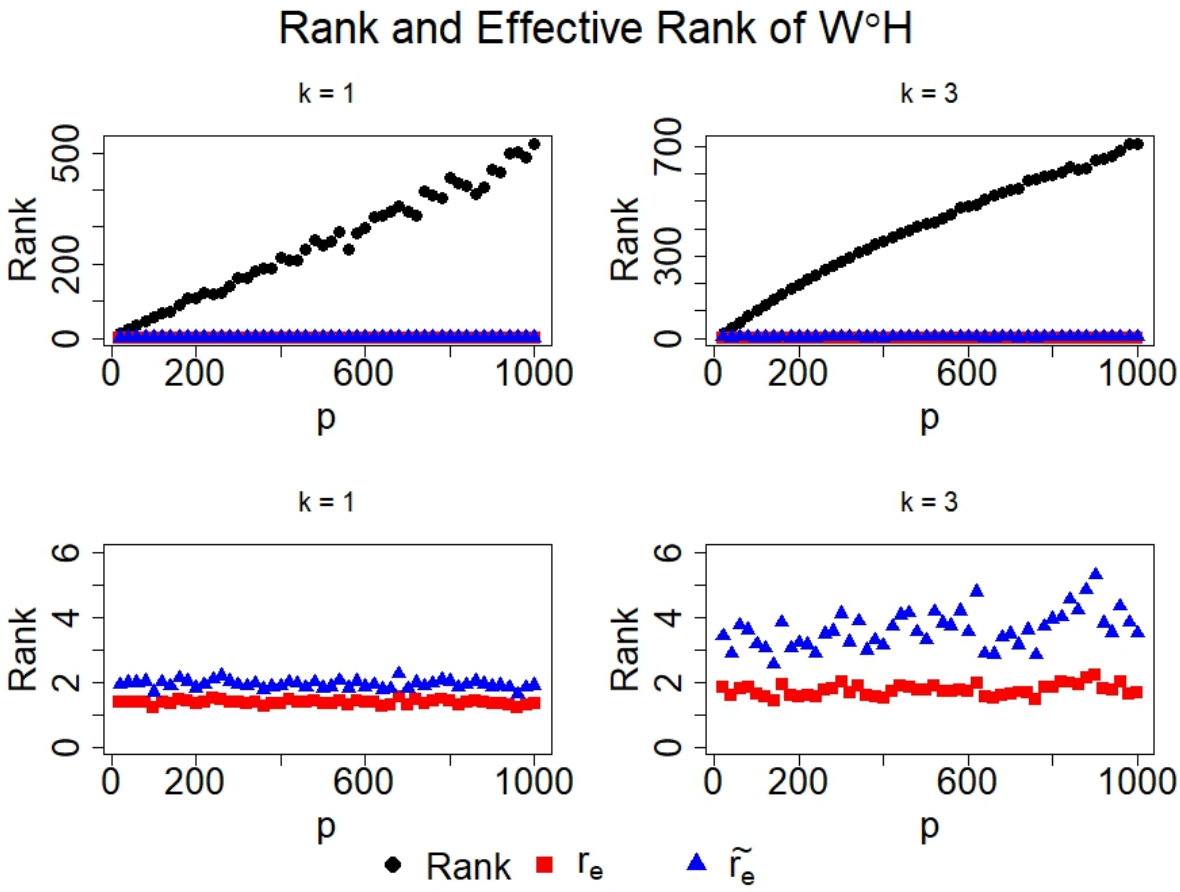

Figure 5.2: The average rank and effective ranks of $W \circ H$ when the $\lambda_j \sim \text{N}(0, 1/2\text{I}_k)$ for $j = 1, \ldots, p$. The top row contains the average rank while the bottom row does not.

5.4 Covariance structure exploration

Since H is a nonlinear transformation on $\Lambda\Lambda^{\mathrm{T}}$, which is low rank matrix, a measure of linear dependence may not be the best measure of dependence for H. Instead, we want to explore the effective rank of $W \circ H$. In order to get an idea of if the effective rank is much different than the rank of $W \circ H$, we calculated $r(\cdot)$, $r_e(\cdot)$, and $\tilde{r}_e(\cdot)$ assuming $\lambda_j \sim \mathrm{N}(0, 1/2\mathrm{I}_k)$ for $j = 1, \ldots, p$. We plot the mean of these values over 20 trials in Figure 5.2 for various values of p and k.

The top row of the plot includes the average rank while the bottom row of the plot does not. We can see that both measures of effective rank are much smaller than the rank, that they remain small even as p grows, and that they grow as k grows. We also see that $\tilde{r}_e(\cdot)$ is greater than $r_e(\cdot)$, which was proved in Proposition 5.2.1.

Now that we have seen numerically that the effective rank remains small, we state and prove an analytical result that the dimension reduction is reflected in the effective rank. Our analytical results are only for $r_e(\cdot)$ and $k = 1$.

Theorem 5.4.1. *In the Poisson model and the Negative Binomial model when $r_j = 1$ for all $j = 1, \ldots, p$, if $\lambda_1, \ldots, \lambda_p \overset{iid}{\sim} \mathcal{N}(0, \epsilon^2)$ when $\epsilon = \sqrt{\log(1.5)}/\left[\sqrt{4\log(2p)} + \sqrt{-4\log(0.1)}\right]$, then with probability greater than 0.85, $W \circ H$ has the effective ranks bounded by the values given in Table 5.1.*

$p =$	50	100	500	1000
$r_e(W \circ H) \leq$	3.065238	3.032884	3.006996	3.003664

Table 5.1: Table of exact bounds on the effective rank of $W \circ H$.

The proof will be given subsection 5.5. Since $W \circ H$ has a low effective rank, using Proposition 5.2.2, $W \circ H - W \circ J$ is low effective rank. Then Ω_Y is the sum of a low effective rank and a diagonal matrix. This parallels the Gaussian factor model, where the covariance matrix is the sum

of a low rank matrix and a diagonal matrix. Thus we see that the dimension reduction from the factor model is reflected in the covariance matrix using the effective rank.

5.5 Proof

In order to prove Theorem 5.4.1, we will re-write the $W \circ H$ matrix as VBV, where B is the Gaussian kernel matrix and V is a diagonal matrix. Then we will bound $r_e(VBV)$ by $r_e(B)$ using Ostrowski's Theorem. Lastly, we bound $r_e(B)$. Before we give the proof, we will talk about the Gaussian kernel.

Let b be the Gaussian kernel function, $b(x, y) = e^{-(x-y)^2/2}$. Then B is the Gaussian kernel matrix such that the i, j^{th} entry of B is $b(\lambda_i, \lambda_j)$.

Mercer's Theorem states that a symmetric kernel K can be re-written as

$$K(x, y) = \sum_{n=0}^{\infty} \nu_n \phi_n(x) \phi_n(y)$$

where ν_n are the eigenvalues and $\phi_n(\cdot)$ are the eigenfunctions that are orthonormal with respect to a function $\rho(\cdot)$ [Fasshauer, 2011].

Then, the Gaussian kernel function, $b(x, y)$, has eigenvalues

$$\begin{aligned}
&\nu_n = e^{-na-b} \text{ where} \\
&a = \log\left[2\alpha^2\left\{1 + \left(1 + \frac{1}{\alpha^2}\right)^{1/2}\right\} + 1\right] \\
&b = \frac{1}{2}\log\left[\frac{\alpha^2}{2}\left\{1 + \left(1 + \frac{1}{\alpha^2}\right)^{1/2}\right\} + \frac{1}{4}\right] - \log(\alpha)
\end{aligned}$$

with eigenfunctions $\phi(\cdot)$ that are orthonormal with respect to the density function of $N\left(0, 1/\left[2\alpha^2\right]\right)$ [Fasshauer, 2011]. The eigenfunctions are not stated, because they are not used in the proof, but Fasshauer [2011] states the explicit expressions for the eigenfunctions.

Proof. First we re-write $W \circ H$ as VBV where B is the Gaussian kernel matrix and V is a diagonal matrix. Remember that $w_i = \exp\{\lambda_i^2/2\}$ and $w = [w_1, \ldots, w_p]^{\mathrm{T}}$. Let $D = \text{diag}\{w\}$, so $DBD =$

H and $DHD = W \circ H$. Let $V = DD = \text{diag}\{\exp\{\lambda_j^2\}\}$. Then $W \circ H = VBV$. We will switch between these two notations throughout this subsection. Since V is a diagonal matrix, it is positive definite and since B is the Gaussian kernel matrix, it is positive definite. Thus, $VBV = W \circ H$ is a positive definite matrix.

Next, we will bound $r_e(VBV)$ by $r_e(B)$. Using Ostrowski's Theorem, $|s_i(VBV/p) - s_i(B/p)| \leq |s_i(B/p)| \, \|V^2 - I\|_2$ [Ipsen, 1998]. Thus

$$s_i(B/p)(1 - \left\|V^2 - I\right\|_2) \leq s_i(VBV/p) \leq s_i(B/p)(1 + \left\|V^2 - I\right\|_2).$$

Since $V^2 - I$ is a diagonal matrix,

$$\left\|V^2 - I\right\|_2 = \max_{1 \leq j \leq p} \left\{ e^{2\lambda_j^2} - 1 \right\}.$$

This is maximized when $|\lambda_j|$ is maximized. Since $\lambda_1, \ldots, \lambda_p \sim \text{iid}\, \mathcal{N}(0, \epsilon^2)$, using the Gaussian Concentration Inequality [Massart, 2007],

$$P\left[\max_{1 \leq j \leq p} |\lambda_j| \geq \epsilon \left\{ \sqrt{2\log(2p)} + \sqrt{-2\log(\delta)} \right\} \right] \leq \delta.$$

So, with probability greater than $1 - \delta$,

$$\left\|V^2 - I\right\|_2 = \max_{1 \leq j \leq p} \left\{ e^{2\lambda_j^2} - 1 \right\} \leq e^{2\epsilon^2 \left\{ \sqrt{2\log(2p)} + \sqrt{-2\log(\delta)} \right\}^2} - 1.$$

Setting $\epsilon = \sqrt{\log(1.5)} / \left\{ \sqrt{4\log(2p)} + \sqrt{-4\log(\delta)} \right\}$, $\|V^2 - I\|_2 \leq 0.5$. Thus, $0.5\, s_i(B/p) \leq s_i(VBV/p) \leq 1.5\, s_i(B/p)$.

The $\text{tr}\,(VBV/p) = \sum_{i=1}^p s_i(VBV/p) \leq 1.5 \sum_{i=1}^p s_i(B/p) = 1.5\, \text{tr}\,(B/p)$. Thus

$$r_e(VBV) = \frac{\text{tr}\,(VBV/p)}{s_1(VBV/p)} \leq 3\, r_e(B).$$

Lastly, we will bound $r_e(B)$. We know that

$$\operatorname{tr}(B/p) = \frac{1}{p}\sum_{i=1}^{p} e^{-(\lambda_i-\lambda_i)^2/2} = 1.$$

Then, using Blanchard et al. [2007], we have that with probability greater than $1 - \exp\{-\xi\}$

$$\begin{aligned} s_1(B/p) &= 1 - \sum_{i=2}^{p} s_i(B/p) \\ &\geq 1 - \left(\frac{2\xi}{p}\sum_{i=2}^{\infty}\nu_i\right)^{1/2} - \frac{\xi}{3p} - \sum_{i=2}^{\infty}\nu_i. \end{aligned} \tag{5.4}$$

This means that $r_e(B)$ is less than or equal to the inverse of the bound in (5.4).

Letting $\delta = 0.1$, $\xi = -\log(0.05)$, and $\epsilon = \sqrt{\log(1.5)}/\left[\sqrt{4\log(2p)} + \sqrt{-4\log(0.1)}\right]$, the bounds on the effective ranks for various values of p are given in Table 5.2. □

$p =$	50	100	500	1000
$r_e(B) \leq$	1.021746	1.010961	1.002332	1.001221
$r_e(VBV) \leq$	3.065238	3.032884	3.006996	3.003664

Table 5.2: Table of exact bounds on the effective rank of $W \circ H$ and B.

5.6 Simulation

We simulated the r_e for $W \circ H$ and B. The λ_j's are independently sampled from a $\mathcal{N}(0, \epsilon^2)$ distribution where $\epsilon = \sqrt{\log(1.5)}/\left[\sqrt{4\log(2p)} + \sqrt{-4\log(0.1)}\right]$. We let $k = [1, 3, 5]$ and p be a sequence from 20 to 1000 increasing by 20 each time. For each combination of k and p, we averaged the effective rank over 20 trials. We also show the distribution of 1000 effective ranks of $W \circ H$ for $k = 1$ and $p = [100, 1000]$.

We see in Figure 5.3 that the effective rank of both B and $W \circ H$ are very close to 1 and that is increases as k increases. The bound for $W \circ H$ is not plotted because it is much larger than the

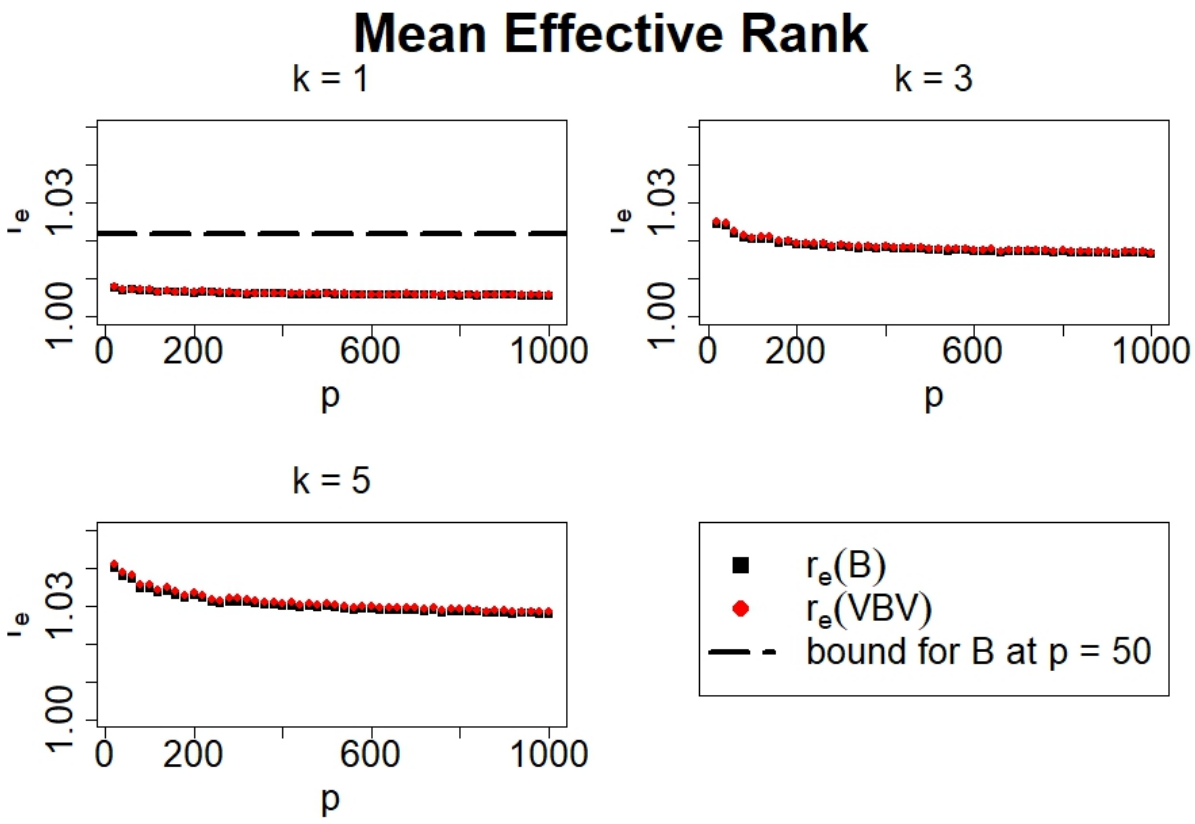

Figure 5.3: Average effective rank over 20 trials for $W \circ H$ and B over various values of p and k.

bound for B. Even though the bound for $W \circ H$ is close to 3 times the bound of B, the actual effective rank of $W \circ H$ is very close to the effective rank of B. Thus, it appears that our bound on $W \circ H$ is a conservative bound.

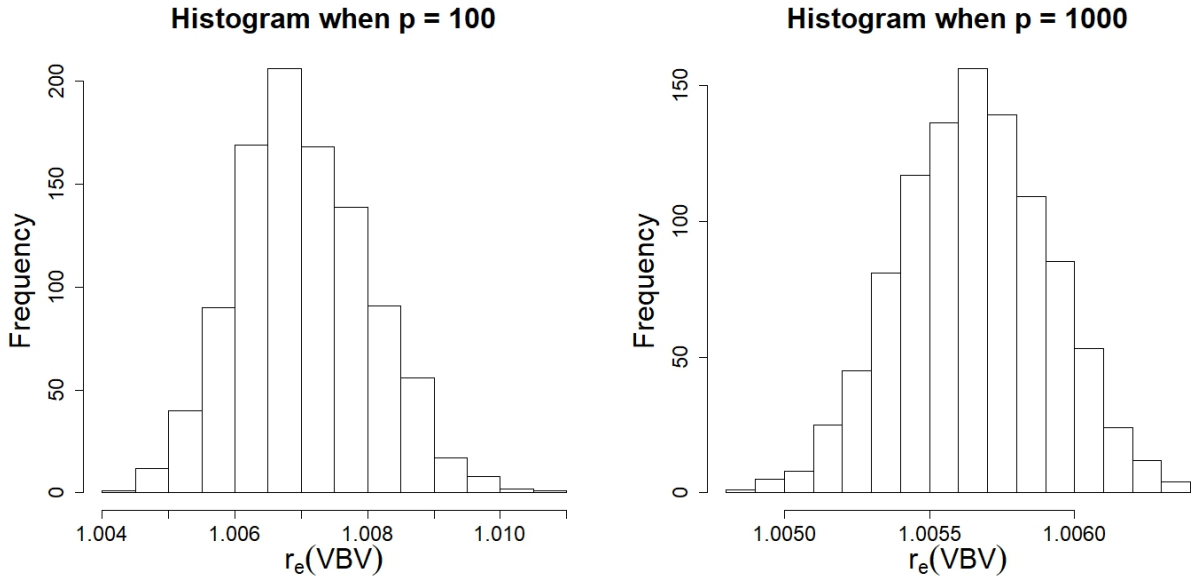

Figure 5.4: Histogram of 1000 draws for $r_e(W \circ H)$ for $k = 1$. The left is plot is when $p = 100$ and the right plot is when $p = 1000$.

The histograms in Figure 5.4 indicate that $r_e(W \circ H)$ is tightly distributed about the mean. The distribution of r_e does not get close to the exact bounds given in Table 5.2, which again indicates that our bound on $W \circ H$ is a conservative bound.

5.7 Discussion

In this section, we explored the covariance structure from a factor model for count data. The sum of a low rank matrix and a diagonal matrix covariance structure of a Gaussian factor model is not seen in the factor models for count data when rank is used as the measure of dependence. However, when effective rank is used as the dependence measure, the covariance matrix has a sum of a low effective rank plus a diagonal matrix structure, which mimics the covariance structure of a

Gaussian factor model. We give the theoretical proof that the covariance matrix of the factor model for count data has a sum of a low effective rank plus a diagonal matrix structure. We also have an exact bound on the low effective rank matrix, which the simulations show is a conservative bound. Future work will focus on making the exact bound more general (for other values of k and larger values of ϵ) and on making the exact bound less conservative.

6. CONCLUSION

We have described two different aMCMC methods, the soft tMVN prior and the SMA regression. The soft tMVN distribution was developed in Section 2 and can be used as an aMCMC method by replacing a usual tMVN prior distribution with the soft tMVN distribution. In Section 3, we developed an aMCMC algorithm for Bayesian shape constrained methods. In it, we used a monotone single-index model and replace the usual tMVN prior with a soft tMVN prior. We showed that the tMVN prior and the soft tMVN prior had similar statistical performance, but the MCMC algorithm using the tMVN prior had almost 5 times the run-time than the aMCMC algorithm using the soft tMVN prior.

In Section 4 we introduce the SMA regression. SMA regression is an aMCMC algorithm for Bayesian convex regression by approximating the maximum of affine functions with the softmax of affine functions. SMA regression is able to perform as well as other leading convex regression methods. While SMA has a longer run-time than various frequentist methods, it allows for immediate quantifications of uncertainty unlike the frequentist methods. While the MCMC method using the usual max of affine functions does not provide run-times, it was unable to scale to more than a few thousand observations and analysis was not performed for more than 1000 observations. SMA, however, was able to scale to at least 10000 observations and the average run-time was less than an hour.

Both aMCMC methods we have described were able to scale better to large datasets than their MCMC counterparts. They were each able to analyze larger datasets and the soft tMVN method had a faster run time. Despite using approximations, they did not lose anything in terms of statistical performance.

REFERENCES

N. E. Aguilera and P. Morin. On convex functions and the finite element method. *SIAM Journal on Numerical Analysis*, 47(4):3139–3157, 2009.

H. Ahn, H. Ichimura, and J. L. Powell. Simple estimators for monotone index models. *manuscript, Department of Economics, UC Berkeley*, 1996.

J. Aitchison and J. Bennett. Polychotomous quantal response by maximum indicant. *Biometrika*, 57(2):253–262, 1970.

J. Aitchison and C. Ho. The multivariate poisson-log normal distribution. *Biometrika*, 76(4): 643–653, 1989.

J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

T. Ando. Bayesian factor analysis with fat-tailed factors and its exact marginal likelihood. *Journal of Multivariate Analysis*, 100(8):1717–1726, 2009.

A. Antoniadis, G. Grégoire, and I. W. McKeague. Bayesian estimation in single-index models. *Statistica Sinica*, pages 1147–1164, 2004.

R. B. Arellano-Valle and A. Azzalini. On the unification of families of skew-normal distributions. *Scandinavian Journal of Statistics*, 33(3):561–574, 2006.

J. Ashford and R. Sowden. Multi-variate probit analysis. *Biometrics*, pages 535–546, 1970.

F. Balabdaoui, C. Durot, and H. Jankowski. Least squares estimation in the monotone single index model. *Bernoulli*, 25(4B):3276–3310, 2019.

R. Bardenet, A. Doucet, and C. Holmes. Towards scaling up markov chain monte carlo: an adaptive subsampling approach. In *Proceedings of the 31st International Conference on Machine Learning*, pages 405–413, 2014.

R. Bardenet, A. Doucet, and C. Holmes. On markov chain monte carlo methods for tall data. *The Journal of Machine Learning Research*, 18(1):1515–1557, 2017.

D. J. Bartholomew. *Latent Variable Models and Factors Analysis*. Oxford University Press, Inc., 1987.

A. Belloni and V. Chernozhukov. High dimensional sparse econometric models: An introduction. In *Inverse Problems and High-Dimensional Estimation*, pages 121–156. Springer, 2011.

A. Bhattacharya and D. B. Dunson. Simplex factor models for multivariate unordered categorical data. *Journal of the American Statistical Association*, 107(497):362–377, 2012.

A. Bhattacharya, A. Chakraborty, and B. K. Mallick. Fast sampling with gaussian scale-mixture priors in high-dimensional regression. *Biometrika*, 103(4):985–991, 2016.

A. Bhattacharya, D. Pati, Y. Yang, et al. Bayesian fractional posteriors. *The Annals of Statistics*, 47(1):39–66, 2019.

G. Blanchard, O. Bousquet, and L. Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 66(2):259–294, 2007.

Z. Botev. The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(1):125–148, 2017.

Z. I. Botev. *TruncatedNormal: Truncated Multivariate Normal*, 2015. URL `https://CRAN.R-project.org/package=TruncatedNormal`. R package version 1.0.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

B. Cai and D. B. Dunson. Bayesian multivariate isotonic regression splines: Applications to carcinogenicity studies. *Journal of the American Statistical Association*, 102(480):1158–1171, 2007.

C. Cavanagh and R. P. Sherman. Rank estimators for monotonic index models. *Journal of Econometrics*, 84(2):351–382, 1998.

P. M. Chak, N. Madras, and B. Smith. Semi-nonparametric estimation with bernstein polynomials. *Economics Letters*, 89(2):153–156, 2005.

Y. Chen and R. J. Samworth. Generalized additive and index models with shape constraints. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(4):729–754, 2016.

S. Chib and E. Greenberg. Analysis of multivariate probit models. *Biometrika*, 85(2):347–361, 1998.

M. Cule, R. Samworth, and M. Stewart. Maximum likelihood estimation of a multi-dimensional log-concave density. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(5):545–607, 2010.

A. S. Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3): 651–676, 2017.

P. Damien and S. G. Walker. Sampling truncated normal, beta, and gamma densities. *Journal of Computational and Graphical Statistics*, 10(2):206–215, 2001.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

D. Dunson. Bayesian latent variable models for clustered mixed outcomes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(2):355–366, 2000.

D. B. Dunson. Dynamic latent trait models for multidimensional longitudinal data. *Journal of the American Statistical Association*, 98(463):555–563, 2003.

D. B. Dunson and B. Neelon. Bayesian inference on order-constrained parameters in generalized linear models. *Biometrics*, 59(2):286–295, 2003.

D. B. Dunson and C. Xing. Nonparametric bayes modeling of multivariate categorical data. *Journal of the American Statistical Association*, 104(487):1042–1051, 2009.

A. Durmus and E. Moulines. High-dimensional bayesian inference via the unadjusted langevin algorithm. *Bernoulli*, 25(4A):2854–2882, 2019.

G. E. Fasshauer. Positive definite kernels: past, present and future. *Dolomite Research Notes on Approximation*, 4:21–63, 2011.

C. Forbes, M. Evans, N. Hastings, and B. Peacock. *Statistical distributions*. John Wiley & Sons, 2011.

J. C. Foster, J. M. Taylor, and B. Nan. Variable selection in monotone single-index models via the adaptive lasso. *Statistics in Medicine*, 32(22):3944–3954, 2013.

A. Frieze and R. Kannan. Log-sobolev inequalities and sampling from log-concave distributions. *The Annals of Applied Probability*, 9(1):14–26, 1999.

A. Frieze, R. Kannan, and N. Polson. Sampling from log-concave distributions. *The Annals of Applied Probability*, pages 812–837, 1994.

J. Geweke. Efficient simulation from the multivariate normal and student-t distributions subject to linear constraints and the evaluation of constraint probabilities. In *Computing science and*

statistics: Proceedings of the 23rd symposium on the interface, pages 571–578. Fairfax, Virginia: Interface Foundation of North America, Inc, 1991.

W. R. Gilks and P. Wild. Adaptive rejection sampling for gibbs sampling. *Applied Statistics*, pages 337–348, 1992.

M. Girolami and S. Rogers. Variational bayesian multinomial probit regression with gaussian process priors. *Neural Computation*, 18(8):1790–1817, 2006.

G. H. Golub and C. F. Van Loan. *Matrix Computations*, volume 3. JHU Press, 2012.

R. B. Gramacy and H. Lian. Gaussian process single-index models as emulators for computer experiments. *Technometrics*, 54(1):30–41, 2012.

P. J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

L. A. Hannah and D. B. Dunson. Bayesian nonparametric multivariate convex regression. *arXiv preprint arXiv:1109.0322*, 2011.

L. A. Hannah and D. B. Dunson. Multivariate convex regression with adaptive partitioning. *The Journal of Machine Learning Research*, 14(1):3261–3294, 2013.

D. J. Henderson and C. F. Parmeter. Imposing economic constraints in nonparametric regression: survey, implementation, and extension. *Advances in Econometrics*, 25(2009):433–69, 2009.

C. Hildreth. Point estimates of ordinates of concave functions. *Journal of the American Statistical Association*, 49(267):598–619, 1954.

C. A. Holloway. On the estimation of convex functions. *Operations Research*, 27(2):401–407, 1979.

C. C. Holmes, L. Held, et al. Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis*, 1(1):145–168, 2006.

R. Horn and C. Johnson. Matrix algebra. *Cambridge University, Cambridge, UK*, 1985.

Z. Huang and A. Gelman. Sampling for bayesian computation with large datasets. *Available at SSRN 1010107*, 2005.

I. C. Ipsen. Relative perturbation results for matrix eigenvalues and singular values. *Acta Numerica*, 7:151–201, 1998.

J. Johndrow, D. Dunson, and K. Lum. Diagonal orthant multinomial probit models. In *Artificial Intelligence and Statistics*, pages 29–38, 2013.

J. Johndrow, J. Mattingly, S. Mukherjee, and D. Dunson. Approximations of markov chains and high-dimensional bayesian inference. *arXiv preprint*, 2015.

J. E. Johndrow, P. Orenstein, and A. Bhattacharya. Bayes shrinkage at gwas scale: Convergence and approximation theory of a scalable mcmc algorithm for the horseshoe prior. *arXiv preprint arXiv:1705.00841*, 2017.

A. Klami, A. Tripathi, J. Sirola, L. Väre, and F. Roulland. Latent feature regression for multivariate count data. In *Artificial Intelligence and Statistics*, pages 462–470, 2015.

A. Korattikara, Y. Chen, and M. Welling. Austerity in mcmc land: Cutting the metropolis-hastings budget. In *International Conference on Machine Learning*, pages 181–189, 2014.

J. H. Kotecha and P. M. Djuric. Gibbs sampling approach for generation of truncated multivariate gaussian random variables. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 3, pages 1757–1760. IEEE, 1999.

J. F. Lawless. Negative binomial and mixed poisson regression. *Canadian Journal of Statistics*, 15 (3):209–225, 1987.

N. D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in neural information processing systems*, pages 329–336, 2004.

S. Lee, P. E. Chugh, H. Shen, R. Eberle, and D. P. Dittmer. Poisson factor models with applications to non-normalized microrna profiling. *Bioinformatics*, 29(9):1105–1111, 2013.

L. Lovász and S. Vempala. Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 57–68. IEEE, 2006a.

L. Lovász and S. Vempala. Simulated annealing in convex bodies and an o*(n^4) volume algorithm. *Journal of Computer and System Sciences*, 72(2):392–417, 2006b.

S. Luo and S. Ghosal. Forward selection and estimation in high dimensional single index models. *Statistical Methodology*, 33:172–179, 2016.

R. Luss, S. Rosset, M. Shahar, et al. Efficient regularized isotonic regression with application to gene–gene interaction search. *The Annals of Applied Statistics*, 6(1):253–283, 2012.

H. Maatouk and X. Bay. Gaussian process emulators for computer experiments with inequality constraints. *Mathematical Geosciences*, 49(5):557–582, 2017.

P. Massart. *Concentration inequalities and model selection*, volume 6. Springer, 2007.

R. L. Matzkin et al. Semiparametric estimation of monotone and concave utility functions for polychotomous choice models. *Econometrica*, 59(5):1315–1327, 1991.

R. Mazumder, A. Choudhury, G. Iyengar, and B. Sen. A computational framework for multivariate convex regression and its variants. *Journal of the American Statistical Association*, 114(525): 318–331, 2019.

R. E. McCulloch, N. G. Polson, and P. E. Rossi. A Bayesian analysis of the multinomial probit model with fully identified parameters. *Journal of Econometrics*, 99(1):173–193, 2000.

G. McLachlan and D. Peel. *Finite Mixture Models*, chapter Mixtures of Factor Analyzers. John Wiley & Sons, Inc., NJ, USA, 2000. doi: 10.1002/0471721182.ch8.

M. C. Meyer, A. J. Hackstadt, and J. A. Hoeting. Bayesian estimation and inference for generalised partial linear models using shape-restricted splines. *Journal of Nonparametric Statistics*, 23(4): 867–884, 2011.

R. F. Meyer and J. W. Pratt. The consistent assessment and fairing of preference functions. *IEEE Transactions on Systems Science and Cybernetics*, 4(3):270–278, 1968.

A. Y. Mitrophanov. Sensitivity and convergence of uniformly ergodic markov chains. *Journal of Applied Probability*, 42(4):1003–1014, 2005.

I. Moustaki and M. Knott. Generalized latent trait models. *Psychometrika*, 65(3):391–411, 2000.

I. Murray, R. Adams, and D. MacKay. Elliptical slice sampling. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 541–548, 2010.

B. Muthen. Latent variable structural equation modeling with categorical data. *Journal of Econometrics*, 22(1-2):43–65, 1983.

R. M. Neal et al. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2 (11):2, 2011.

W. Neiswanger, C. Wang, and E. Xing. Asymptotically exact, embarrassingly parallel mcmc. *arXiv preprint arXiv:1311.4780*, 2013.

S. M. O'brien and D. B. Dunson. Bayesian multivariate logistic regression. *Biometrics*, 60(3): 739–746, 2004.

Y. Ochi and R. L. Prentice. Likelihood inference in a correlated probit regression model. *Biometrika*, 71(3):531–543, 1984.

A. Pajor et al. Estimating the marginal likelihood using the arithmetic mean identity. *Bayesian Analysis*, 12(1):261–287, 2017.

A. Pakman and L. Paninski. Exact hamiltonian monte carlo for truncated multivariate gaussians. *Journal of Computational and Graphical Statistics*, 23(2):518–542, 2014.

R. D. Payne and B. K. Mallick. Two-stage metropolis-hastings for tall data. *Journal of Classification*, 35(1):29–51, 2018.

N. S. Pillai and A. Smith. Ergodicity of approximate mcmc chains with applications to large data sets. *arXiv preprint arXiv:1405.0182*, 2014.

W. Polasek and A. Krause. The hierarchical tobit model: A case study in bayesian computing. *Operations-Research-Spektrum*, 16(2):145–154, 1994.

N. G. Polson and J. G. Scott. Shrink globally, act locally: Sparse bayesian regularization and prediction. *Bayesian Statistics*, 9:501–538, 2010.

N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349, 2013.

M. Quiroz, R. Kohn, M. Villani, and M.-N. Tran. Speeding up mcmc by efficient data subsampling. *Journal of the American Statistical Association*, 114(526):831–843, 2019.

C. E. Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.

G. O. Roberts and J. S. Rosenthal. Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1): 255–268, 1998.

G. Rodriguez-Yam, R. A. Davis, and L. L. Scharf. Efficient gibbs sampling of truncated multivariate normal with application to constrained linear regression. *Unpublished manuscript*, 2004.

O. Roy and M. Vetterli. The effective rank: A measure of effective dimensionality. In *Signal Processing Conference, 2007 15th European*, pages 606–610. IEEE, 2007.

S. Roy, W. Chen, C. C.-P. Chen, and Y. H. Hu. Numerically convex forms and their application in gate sizing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(9):1637–1647, 2007.

D. Rudolf, N. Schweizer, et al. Perturbation theory for markov chains via wasserstein distance. *Bernoulli*, 24(4A):2610–2639, 2018.

M. Sammel, L. Ryan, and J. Legler. Latent variable models for mixed discrete and continuous outcomes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(3): 667–678, 1997.

M. J. Schell and B. Singh. The reduced monotonic regression method. *Journal of the American Statistical Association*, 92(437):128–135, 1997.

S. L. Scott, A. W. Blocker, F. V. Bonassi, H. A. Chipman, E. I. George, and R. E. McCulloch. Bayes and big data: The consensus monte carlo algorithm. *International Journal of Management Science and Engineering Management*, 11(2):78–88, 2016.

T. S. Shively, S. G. Walker, and P. Damien. Nonparametric function estimation subject to monotonicity, convexity and other shape constraints. *Journal of Econometrics*, 161(2):166–181, 2011.

A. Skiba. Optimal growth with a convex-concave production function. *Econometrica: Journal of the Econometric Society*, pages 527–539, 1978.

A. Souris, A. Bhattacharya, and D. Pati. The soft multivariate truncated normal distribution. *arXiv preprint arXiv:1807.09155*, 2018.

G. P. Styan. Hadamard products and multivariate statistical analysis. *Linear Algebra and its Applications*, 6:217–240, 1973.

S. D. Team et al. Stan modeling language: User's guide and reference manual, 2016.

J. Tobin. Estimation of relationships for limited dependent variables. *Econometrica: Journal of the Econometric Society*, pages 24–36, 1958.

H. R. Varian. The nonparametric approach to production analysis. *Econometrica: Journal of the Econometric Society*, pages 579–597, 1984.

R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

C. Villani. *Optimal Transport: Old and New*, volume 338. Springer Science & Business Media, 2008.

H.-B. Wang. Bayesian estimation and variable selection for single index models. *Computational Statistics & Data Analysis*, 53(7):2617–2627, 2009.

J. Wang and S. K. Ghosh. Shape restricted nonparametric regression with bernstein polynomials. *Computational Statistics & Data Analysis*, 56(9):2729–2741, 2012.

X. Wang and D. B. Dunson. Parallelizing mcmc via weierstrass sampler. *arXiv preprint arXiv:1312.4605*, 2013.

M. Wedel, U. Böckenholt, and W. A. Kamakura. Factor models for multivariate count data. *Journal of Multivariate Analysis*, 87(2):356–369, 2003.

M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

M. West. Bayesian factor regression models in the "large p, small n" paradigm. *Bayesian Statistics*, 7:723–732, 2003.

S. Wilhelm and M. B. G. *tmvtnorm: Truncated Multivariate Normal and Student t Distribution*, 2015. URL `http://CRAN.R-project.org/package=tmvtnorm`. R package version 1.4-10.

Y. Yu and D. Ruppert. Penalized spline estimation for partially linear single-index models. *Journal of the American Statistical Association*, 97(460):1042–1054, 2002.

X. Zhang, W. J. Boscardin, and T. R. Belin. Bayesian analysis of multivariate nominal measures using multivariate multinomial probit models. *Computational Statistics & Data Analysis*, 52(7): 3697–3708, 2008.

J. Zhou, A. Bhattacharya, A. H. Herring, and D. B. Dunson. Bayesian factorizations of big sparse tensors. *Journal of the American Statistical Association*, 110(512):1562–1576, 2015.

M. Zhou and L. Carin. Negative binomial process count and mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):307–320, 2015.

M. Zhou, L. Hannah, D. Dunson, and L. Carin. Beta-negative binomial process and poisson factor analysis. In *Artificial Intelligence and Statistics*, pages 1462–1471, 2012a.

M. Zhou, L. Li, D. Dunson, and L. Carin. Lognormal and gamma mixed negative binomial regression. In *Proceedings of the International Conference on Machine Learning. International Conference on Machine Learning*, volume 2012, page 1343. NIH Public Access, 2012b.

S. Zhou, P. Giulani, J. Piekarewicz, A. Bhattacharya, and D. Pati. Reexamining the proton-radius problem using constrained gaussian processes. *Phys. Rev. C*, 99:055202, May 2019a. doi: 10.1103/PhysRevC.99.055202. URL https://link.aps.org/doi/10.1103/PhysRevC.99.055202.

S. Zhou, P. Giulani, J. Piekarewicz, A. Bhattacharya, and D. Pati. Reexamining the proton-radius problem using constrained gaussian processes. *Physical Review C*, 99(5):055202, 2019b.

APPENDIX A

A.1 Proof of Proposition 2.2.1

Let $\gamma(\theta) = g(\theta)/M$, where

$$g(\theta) = e^{-\frac{1}{2}(\theta-\mu)'\Sigma^{-1}(\theta-\mu)} \prod_{i\in[r]\,:\,s_i=1} \mathbb{1}(a_i'\theta \geq 0) \prod_{i\in[r]\,:\,s_i=-1} \mathbb{1}(a_i'\theta < 0),$$

and $M = \int_{\mathbb{R}^d} g(\theta)d\theta$ is the normalizing constant. Similarly, let $\gamma_\eta(\theta) = g_\eta(\theta)/M_\eta$, where

$$g_\eta(\theta) = e^{-\frac{1}{2}(\theta-\mu)'\Sigma^{-1}(\theta-\mu)} \prod_{i\in[r]\,:\,s_i=1} \sigma_\eta(a_i'\theta) \prod_{i\in[r]\,:\,s_i=-1} \{1-\sigma_\eta(a_i'\theta)\},$$

where recall $\sigma_\eta(\cdot)$ is the sigmoidal function. Bound

$$\begin{aligned}
&\int_{\mathbb{R}^d} |\gamma(\theta) - \gamma_\eta(\theta)|d\theta \\
&\leq M^{-1} \int |g(\theta) - g_\eta(\theta)|d\theta + |1/M - 1/M_\eta| \int g_\eta(\theta)d\theta \\
&\leq M^{-1}\Big[\int |g(\theta) - g_\eta(\theta)|d\theta + |M - M_\eta|\Big],
\end{aligned}$$

where we have used triangle inequality and the fact that $\int g_\eta(\theta)d\theta = M_\eta$. Now, we have

$$|M - M_\eta| = \Big|\int (g(\theta) - g_\eta(\theta)d\theta\Big| \leq \int |g(\theta) - g_\eta(\theta)|d\theta.$$

Thus, we have

$$\int_{\mathbb{R}^d} |\gamma(\theta) - \gamma_\eta(\theta)|d\theta \leq 2M^{-1} \int |g(\theta) - g_\eta(\theta)|d\theta. \tag{A.1}$$

Now, using that

$$|\mathbb{1}(a_i'\theta \geq 0) - \sigma_\eta(a_i'\theta)| = |\mathbb{1}(a_i'\theta < 0) - \{1 - \sigma_\eta(a_i'\theta)\}| = \frac{1}{1 + e^{\eta|a_i'\theta|}}$$

and the fact that for numbers $u_i, v_i \in [0, 1]$,

$$\left| \prod_{i=1}^{r} u_i - \prod_{i=1}^{r} v_i \right| \leq \sum_{i=1}^{r} |u_i - v_i|,$$

we have that

$$\int |g(\theta) - g_\eta(\theta)| d\theta \lesssim \sum_{i=1}^{r} E\left[\frac{1}{1 + e^{\eta|a_i'\theta|}}\right],$$

where $a \lesssim b$ means $a \leq Cb$ for some positive constant C, and the expectation E is under a $\mathcal{N}(\mu, \Sigma)$ distribution. By monotone convergence theorem, the right hand side of the above display converges to 0 as $\eta \to \infty$.

APPENDIX B

B.1 Monotone Single Index Model Results when $\eta = 100$

The top two plots in figure B.1 shows the pointwise posterior mean of f depending on two different initializations, setting $\eta = 100$. The first realization (left) is from both Gibbs samplers with a starting value for β as a vector of 1's. The second realization (right) is from both Gibbs samplers with a starting value for β as a vector of -1's. For both starting values, the Gibbs sampler with a tMVN prior is able to estimate the true function well. However, the Gibbs sampler with a soft tMVN prior is sensitive to the starting value of β and the value of η. Since $\sum_{j=0}^{M} \theta_j \tilde{B}_{M,j}(X^{\mathrm{T}}\alpha)$ assuming that θ is non-decreasing is equivalent to $\sum_{j=0}^{M} -\theta_j \tilde{B}_{M,j}\{X^{\mathrm{T}}(-\alpha)\}$ assuming that θ is non-increasing and the soft tMVN does not enforce a hard constraint, the soft tMVN algorithm is trying to estimate the function with the sign flip. This is easy to spot, and since the soft tMVN algorithm runs quickly, it is easy to try multiple different starting values for β and different values of η, and choose a value of η so that the sampler is not sensitive to the starting value of β. The bottom plot of figure B.1 shows the pointwise posterior mean of f using random starting point for β and setting $\eta = 500$. Both Gibbs samplers are able to estimate the true function well.

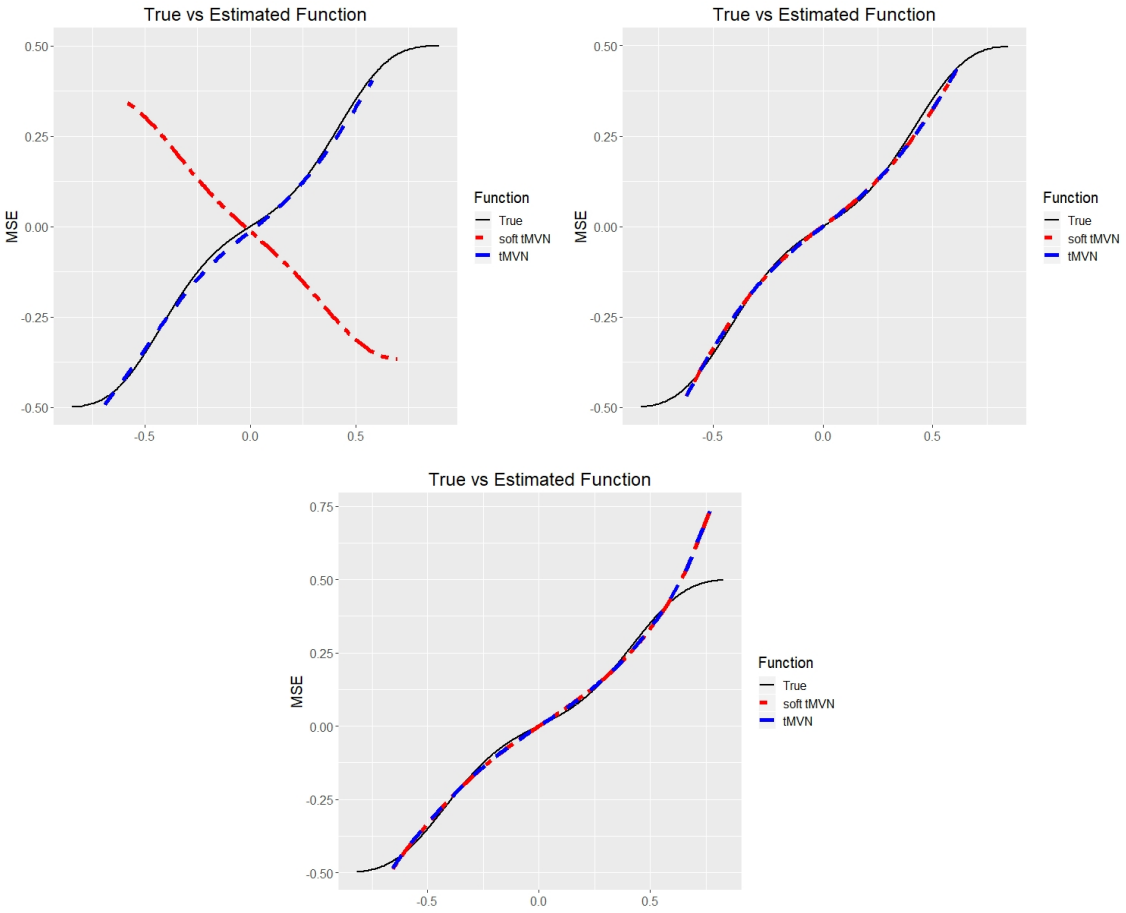

Figure B.1: Plots of the true function (solid black line) and estimated functions using the soft tMVN (red dot dashed line) and the tMVN (blue dashed line) priors. The left plot assumes the starting value for β is a vector of 1's and the right lot assumes the starting value for β is a vector of -1's. The top two plots have η set at 100. The bottom plot assumes random starting values for β and has η set at 500.